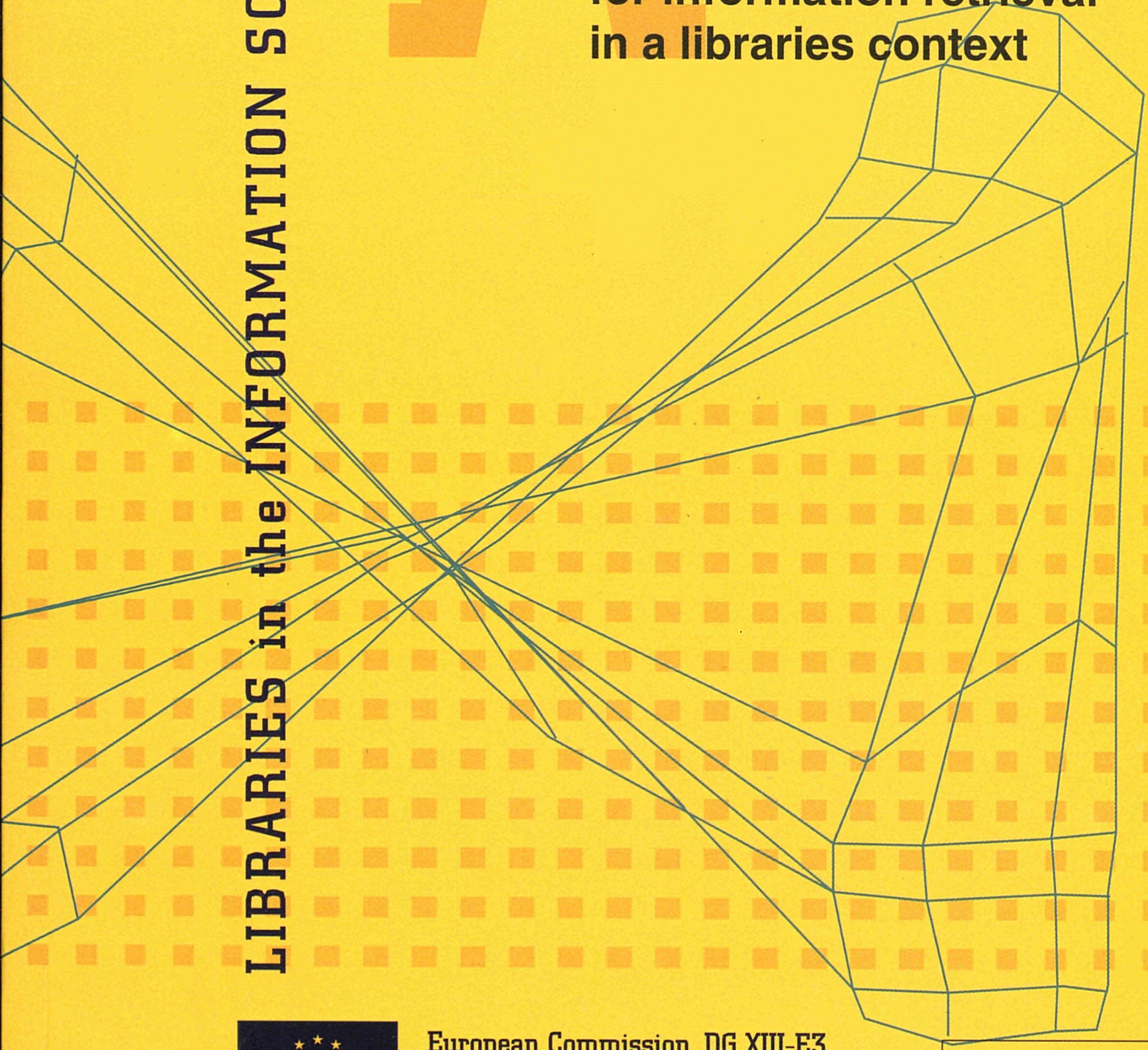
ISSN 1018-5593

1995

LIBRARIES in the INFORMATION SOCIETY

Artificial neural networks for information retrieval in a libraries context



European Commission, DG XIII-E3

EUR 16264 EN

1995

LIBRARIES in the INFORMATION SOCIETY

Artificial neural networks for information retrieval in a libraries context

Author:
Dr ir Johannes C. Scholtes

European Commission, DG XIII-E3

EUR 16264 EN

Published by the
EUROPEAN COMMISSION

Directorate-General XIII
Telecommunications, Information Market and Exploitation of Research

L-2920 Luxembourg

LEGAL NOTICE

Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use which might be made of the following information

Cataloguing data can be found at the end of this publication

Luxembourg: Office for Official Publications of the European Communities, 1995

ISBN 92-827-4690-9

Printed in Italy

Artificial Neural Networks for Information Retrieval in a Libraries Context

Executive Summary

Background

From February 1994 up to September 1994, M.S.C. Information Retrieval Technologies BV, based in Amsterdam, the Netherlands, undertook a study on *Neural Networks and Information Retrieval in a Libraries Context,* in co-operation with the Department of Computational Linguistics of the University of Amsterdam and the Department of Information Technology and Information Science at Amsterdam Polytechnic. This study is funded by the European Commission, DG XIII, as an accompanying measure under the Libraries Programme[1].

So far, the European Commission has funded over 40 projects of different sizes under the ESPRIT programme and other programmes which involve research on or the application of artificial neural network (ANN) technology.

Despite the theoretical and practical evidence that ANN are good tools for pattern recognition tasks, it was still an open question whether they were appropriate tools within the specific domain of Bibliographic Information Retrieval. Apart from some minor studies no real attempt has been made to integrate an ANN as a main component of a bibliographical information retrieval system or an on-line public access catalogue (OPAC). It was therefore not clear whether and how ANN techniques could be combined with more "classical" methods, for instance rule-based or statistical approaches. By the same token it was not clear either to what extent existing OPAC's could benefit from ANN technology.

Objectives

The objectives of this study were:

[1] Libraries are one of the seven areas covered by the specific subprogramme "Telematics in Areas of General Interest" within the "3rd Framework Programme on Research and Technology Development" of the European Union.

(i) to ascertain the state-of-the-art of the application of ANN technology to Information Retrieval (IR), with particular emphasis on bibliographic information in a libraries context;

(ii) to assess the (potential) quality of ANN-based approaches to IR in this particular domain of interest, in comparison with traditional practices. Here "quality" must be understood in terms of both (measurable) efficiency and practical benefits;

(iii) to stimulate interest in the practical application of ANN technology to bibliographic information retrieval in a libraries context.

In order to discuss and disseminate the results obtained through this study, two one-day workshops have been organised by M.S.C. Information Retrieval Technologies BV, the first one after compilation of the State-of-the-art Report and the second one after completion of the prototyping and experimentation phase.

State-of-the-art Report

The subject of investigation in the state-of-the-art study was the general application of ANN technology to IR problems in a libraries context. Typical applications of this technology are advanced interface design, current awareness, Selective Dissemination of Information (SDI), fuzzy search and concept formation on bibliographical databases as well as on full-text documents.

The state-of-the-art report indicated over 300 directly and 200 in-directly related references on the application of ANN's in IR. Most of them showed interesting results on "toy problems", but hardly any of the studies showed real improvements over existing IR technology on large databases as they are used in a library.

Prototypes

As a result of the state-of-the-art report and the first workshop, a number of prototypes were developed. The target applications of the prototypes were highly inspired by the possible level of success one could expect. The directions chosen were:

1. A fuzzy extension to a traditional information retrieval algorithm in order to retrieve information from corrupted texts such as optical character recognition (OCR) documents. Here the "fuzzy mapping" between words and their OCR variants was trained to an ANN.

2. An information filter that was able to retrieve parts of large information flows according to some “easy to define and maintain user model”. In this application the user model was stored in an ANN.

3. A bibliographical browser that would allow one to jump between clusters of related documents. These clusters were derived automatically by ANN’s.

These three ANN extensions were compared to traditional IR techniques that had proven to be successful.

Results and Conclusions

In all of these applications, extensive comparisons showed that ANN’s can hardly do better on “library data” than traditional approaches. In addition, none of the workshop participants could provide suggestions for improvements of the prototypes that would directly lead to outperforming of the traditional models. The only suggestions done were hints for future research.

The main reasons for this failure of ANN’s in performance were:

- A too large dimensionality of the typical library data sets. As a result the ANN’s no longer converged to proper end-states. One should be aware of the fact that success of ANN’s in toy problems does not guarantee success on larger data sets.
- Required computational power. The size of typical library data sets is too large to be processed by ANN’s. It just takes too long.
- A typical ANN is suited for processing noisy “natural” signals such as sound or vision. Bibliographical records do not contain such data, therefore such an application does not use the typical abilities of ANN’s and is often an overkill.
- ANN usually do not make assumptions about the characteristics of a particular data set. This makes them the “second best solution to any problem” and justifies their use in situations where the characteristics of the data are unknown or where a quick-and-dirty solution must be found fast. However for mission critical applications in libraries specialised (optimal) algorithms remain advisable.

In general, the information stored in libraries does not contain any data that is especially suited for processing by ANN’s or that could take particular advantages of ANN’s.

Nevertheless, it might be worthwhile to apply ANN's to certain tasks in a more general "information engineering" setting, because sometime one just doesn't understand the problems or the data that well, or one doesn't have the time to understand the problem. In these cases the neural network metaphor can provide useful insights.

But, when applying ANN's, one should be aware to use them for applications that take advantage of typical ANN properties. Some guidelines are:

- derive non-linear mappings of badly defined problems by training on collected examples,
- keep the dimensionality of the problem small (sometimes this can be done by data compression),
- work on data that are as "natural" as possible,
- work on noisy data sets,
- be restricted to a very limited time-frame in order to solve the problem.
- keep in mind that ANN's are not an *enabling technology*, but at the most an *enabling metaphor*
- use sound methodology to evaluate the performance of ANN's, and compare this to traditional statistical pattern recogntion techniques on the same task.

Amsterdam,, March 28th, 1995

Dr Johannes C. Scholtes, M.S.C. Information Retrieval Technologies BV

Tal·e of Contents

Preface

This report has been written by M.S.C. Information Retrieval Technologies B.V., based in Amsterdam, the Netherlands, in collaboration with the Department of Computational Linguistics of the University of Amsterdam and the Department of Information Technology and Information Science at Amsterdam Polytechnic. It is part of a study into the possibilities of the application of Artificial Neural Networks (ANN) for Information Retrieval (IR) in a Libraries Context. This study is funded by the European Commission, DG XIII, as an accompanying measure under the Libraries Programme (contract number PROLIB/ANN).

Part of this report has been used as working notes for the first workshop "Neural Networks for Information Retrieval in a Libraries Context", organised by M.S.C. in Amsterdam, June 24, 1994. At this workshop the State-of-the-art and the exact directions of the prototyping phase were discussed.

A demonstration of the prototypes and the results of the investigations have been presented at the workshop organised by MSC in Amsterdam, September 16, 1994, followed by the delivery of this final report in October 1994.

This report appears in this form due to the valuable suggestions of Edwin Brinkhuis, Hans Henseler, Anita Lettink, Micha Leuw, Remko Scha, Eric Sieverts, Marco-René Spruit, Hans-Georg Stork, Jakub Zavrel and Henk Zeevat.

Introduction

Recent research of artificial neural networks (ANN) in the field of pattern recognition and pattern classification applications has provided successful alternatives to traditional techniques. Products applied for optical character recognition (OCR), speech recognition, hand-written character recognition and prediction of non-linear time series are good examples of the commercialisation of ANN techniques. So far, the European Commission has funded over 40 projects of different sizes under the ESPRIT and other programmes which involve research on or the application of ANN technology.

The task of Information Retrieval (IR), i.e. the matching of a large number of documents against a query, can also been seen as a pattern recognition or pattern classification task. There have been several approaches to the application of ANN in IR in order to increase the quality of the retrieval process.

Despite the theoretical and practical evidence that ANN's are good tools for pattern recognition tasks, it was still an open question whether they are appropriate tools within the specific domain of Bibliographic Information Retrieval. Apart from some minor studies it seems that until now no real attempt has been made to integrate an ANN as a main component of a bibliographical information retrieval system or an on-line public access catalogue (OPAC). It is therefore not clear whether and how ANN techniques can be combined with more "classical" methods, for instance rule-based or statistical approaches. It is not clear either to what extent existing OPAC's could benefit from ANN technology.

Traditionally, in a libraries context one has to cope with applications such as categorisation of bibliographic data, information localisation, information retrieval, loan and serial management and the acquisition of new titles.

But libraries are changing:

- They move from a traditional role of information archival towards the role of information signalling and distribution (previously a typical publisher's task),
- more and more data is digitally available, and
- the character of information that is archived by libraries is changing from pure text towards different media types such as pictures, video and sound.

In this report, these shifts in activities are taken into account. Current as well as future library tasks are studied with respect to ANN's.

Neural techniques have shown considerable success for unstructured, incomplete, "noisy" or fuzzy data sets. In comparison to traditional (e.g. rule-based) techniques ANN's have been shown to be more robust when applied with such data. Most of those typical pattern recognition applications were in low-level signal processing such as robot-arm movements or image processing. Here it is investigated if these neural techniques can also be used for particular higher level tasks such as information retrieval.

In information retrieval a large collection of natural language data (structured or unstructured) is made accessible by a computer system, which must provide the user with relevant text passages or records according to some query. The main problems in this application are caused by the fact that natural language displays ambiguities on all levels, leading to an enormous number of possible interpretations in retrieval. In one way or another, human beings are capable of processing language accurately and efficiently without too much effort. Thus, the study toward more biologically inspired computer models in information retrieval tasks, such as ANN's, seems plausible.

A "natural" application of neural networks in information retrieval is the automatic organising or clustering of information. In a libraries context this can be seen as the generation of thesauri or the automatic categorisation of bibliographic information. This concept can quit easily be extended to searching in multimedia information such as pictures, video or sound, as these are always noisy data sets. In addition many smaller sub-tasks of information retrieval in libraries can be identified, where ANN's might be of use.

The objectives of this study are:

- to ascertain the State-of-the-Art of the application of Artificial Neural Net (ANN) technology to Information Retrieval (IR), with particular emphasis on bibliographic information in a libraries context;
- to assess the (potential) quality of ANN-based approaches to IR in this particular domain of interest, in comparison with traditional practices. Here "quality" must be understood in terms of both (measurable) efficiency and practical benefits;
- to stimulate interest in the practical application of ANN technology to bibliographic information retrieval in a libraries context.

This report consists of four parts. Part One is an introduction to the research. It is intended to provide background knowledge about libraries, information retrieval and neural networks at a level necessary to understand the findings of the state of the art report. In particular, a discussion is given of areas in libraries and in information retrieval where ANN's seem applicable. The types of suitable neural network architectures are exposed in detail, and an overview is given of the general ideas behind the neural computation approach.

Part Two describes the state-of the-art. It is the result of an extensive literature study into the application of ANN's in IR. Here existing applications of neural net technology in areas such as clustering, thesaurus construction, interface design, hypertext generation, adaptive databases, query generalisation, database mining, advanced help desk systems, user modelling in current awareness and selective dissemination of information, serial and loan management, acquisition of new titles or juke-box staging, as found in the literature are described. These systems are more or less successful, depending on various factors. It is argued why certain applications are better than others, why some applications will and why some others will never work. When available, commercial applications are discussed.

In Part Three the three prototypes which were a part of the commissioned study are described. These include a fuzzy search prototype, a document clustering prototype and an information filtering prototype. An explanation is given why these particular areas were chosen., details of their functioning. In the case that the prototypes involve extensions to the state-of-the-art, the particular technical information is given. In each case the utility of the ANN approach is discussed and an empirical comparison is made of the performance of the prototypes compared to more traditional techniques.

Part Four contains the general discussion and the conclusions. It discusses the lessons learned from the literature review and from the analysis of the prototypes, and gives practical guidelines and recommendations. In the end, a general framework is given that can be useful for the determination of the potential success of an ANN application in information retrieval.

Part 1

Background

1 Changes in Libraries

παντα ρει, νου μενει

-- *Heraclitus*

In this chapter, some issues with respect to the libraries context are discussed. Although it is not the exact scope of this project, it is regarded as important in order to be able to asses the impact of neural net technology on information retrieval in a library setting.

There are currently a number of important issues which libraries must face. Libraries are going through rapid changes, and it might not be exaggerated to speak of an information revolution. For many of the problems that accompany these changes librarians are hoping to get assistance from technological progress. Some of these issues are:

- The limited accessibility of information to the general public.
- The shift in role from information archive towards information distributor.
- The change of the character of the information from text-only towards different types of media such as sound, pictures and video.
- The amount of information, which is growing exponentially.
- The change of information storage from paper and micro-film based (or analogue) information sources towards digital information sources such as CD ROM's, on-line services and computer networks.

Limited Accessibility

Traditionally, libraries have been archiving textual information. In the process of doing so, information had to be categorised according to some scheme. Based on this scheme information could be localised and retrieved. As computers got more powerful, it was possible to store the library index in a computer system, making it faster and easier accessible. Much of the information that has been stored in such a way was highly structured in records and fields such as title, author, publisher, subject, etc.

One of the main problems in accessing information by means of structured records is caused by the fact that one cannot classify information consistently into manually designed categories (different people use different interpretations, even the interpretation of one person varies over time!). In addition, the process of structuring is very expensive and causes a delay in time. Therefore, and because nothing represents the information better than the information itself, full-text (search methods that work on every individual word in the information) access is necessary as an additional search method. In a practical libraries context, this need has been translated to the full-text availability of (hand made) abstracts.

However, full-text access as it is used now, has a number of limitations:

- It requires the user to have some knowledge of the contents of the information store beforehand, because keywords must be entered on the fly.
- If search terms are misspelled or if the information is noisy in other ways, it cannot be located easily.
- One is never certain if every relevant piece of information has been located.
- In many cases, too much non-relevant information is retrieved.

Searching in bibliographic information is therefore often the task of a specialist, as a result of which the information in a library is not easily accessible to the general public without professional assistance. This situation could be greatly improved by the addition of more effective full-text retrieval capabilities to the on-line public access catalogue (OPAC). At the same time care must be taken to equip such retrieval tools with intuitively easily understandable interfaces.

From Information Archive Towards Information Distributor

One of the most interesting changes in the nature of a library is the shift from information archive towards information signaller and distributor. In the past, distribution was mainly the task of a publisher. However, more and more libraries have started similar services for:

- Current awareness
- Selective dissemination of information (SDI)
- Distribution of information

It is here where the need for sensitivity to the users' context is the highest, because traditional techniques provide either no information or too much information. User modelling is a highly complex task in which there is a major role for adaptivity (the automatic change of the interest model). In most cases these tasks are either implemented manually or by global surface analysis of the information.

From Text-Only Towards Multi-Media

In the past, information was only available in text. At this moment more and more libraries archive sound-recordings, pictures (photography, paintings, drawings and posters), and video. For the same reason as why one needs full-text access in a text-only environment, one might consider full multi-media access in a multi-media environment. This means that one would like to search through pictures by providing the system with a picture-like query instead of searching through manually added textual labels.

This need can be seen as a completely new dimension in information access for which no standards exist today. There is on-going basic research towards searching in multi-media, but this is all very premature. Although this area is not the focus of the present study, one should assure that new information retrieval technologies are not prima facie incompatible with the developments in multi-media.

From Megabytes Towards Terabytes of Information

Over the years, the amount of information that has to be stored in a library has grown exponentially. In the first place, the traditional collection expands continually. The number of published titles increases heavily every year. In the second place the amount of information stored in the library electronically is exploding in recent years. This is because the new types of information (multi-media) require Mega and Gigabytes of storage. At the same time, the information that has to be stored per title, regardless of media type, is increasing (from structured information only, towards the addition of abstracts and even the full text),

As a result, the storage capacity and search power needed to locate, retrieve, and browse this information have grown significantly.

From Analogue Information Sources Towards Digital Information Sources

Where information used to be analogue only (paper, micro-film, photos), more and more information is available in digital form due to:

- On-line services
- CD-technology

- Computer networks
- Larger storage capacity

As information is digitally available, one can access the full information easier. In the past, full-text searching of a scientific paper was only possible after an expensive Optical Character Recognition (OCR) process.

If the information is provided digitally, it becomes much easier to store information, search in the it full-text, filter the information and distribute it.

The Future of Libraries

Given the above mentioned items, libraries need to adjust to these new tasks, new information volumes and new media. In order to do so, they need new software to assist them in carrying out both their traditional and their new tasks. It is here where advanced mathematical models, such as artificial neural networks might be of help.

2 Information Retrieval

"The only good data is more data"

-- Robert Mercer

In this chapter, the study of information retrieval is discussed in a more thorough manner. Besides a brief introduction of the used techniques and evaluation methods, a short list of the major problems is given.

It can be stated that Information Retrieval (IR) is the ultimate combination between Natural Language Processing (NLP) and Artificial Intelligence (AI). On the one hand there is an enormous amount of natural language data that needs to be processed and understood to return the proper information to the user. On the other hand, one needs to understand what the user intends with his or her query given the context of the other queries and some kind of user model.

2.1 Introduction

There are a number of methods in information retrieval that are difficult to beat.

- Adjacent character statistics or n-gram analyses. A window of size n is shifted over a text. For every possible character combination in a stored document, the frequency is kept. By comparing the frequency vector of a query to that of all documents in the data base, a measure of correlation can be calculated [Forney, 1973] [Hull et al., 1982] [D'Amore et al., 1988] [Kimbrell, 1988]. The main advantages of this method are highly robust behaviour and the elimination of a dictionary. As a result, only a global surface analysis of the text is obtained.

- Inverted indices (representing the exact position of every content word in a stored document) are a well known, accurate and fast technique [Sparck Jones, 1971]. Retrieval with inverted indices can be extended to Boolean queries (A and B) and adjacent queries (A within 5 words of B). The difference between single or multiple occurrences in one document is not measured; therefore, ranking the retrieved documents on basis of their relevance is not possible.

- Next, by giving weight values to the index terms, a so-called relevance ranking algorithm can be designed. Hereby, the relevance value of a document is calculated by multiplying the total occurrence of an index term in a document by a relevance factor. Normally, frequently occurring words have low relevance factors, infrequently occurring words have high relevance factors. The documents are ranked in order of their total relevance value. The weights can then be determined manually or automatically. In general, these weights are normalised with respect to the document size [Cooper, 1971].

- Once a certain query is processed, the user can feed the results of a query back into the system. If the user indicates how good or how bad a certain result was, the computer can change the results according to the users input. This technique is called relevance feedback [Salton, 1989]; it is known to be very effective.
- Instead of using the original word in the index, one can use a subset of artificial (semantic) entities. Each natural word is categorised into one semantical group. The index term only contains the semantical notions: latent semantic indexes [Dumais et al., 1988]. This type of models can be extended to conceptual instead of semantical items, called conceptual information retrieval [Mauldin, 1991]. Both techniques can best be seen as an addition to standard inverted (weighted) indices. Due to its manual nature, the maintenance of semantical groups (or concepts) is expensive. Moreover, semantical groups are subject to personal preferences and once a word is categorised into a certain group, it can no longer be retrieved from others.
- Quite useful is the addition of a (layered) thesaurus to the system in order to provide the user with (semantical) alternatives for key words in his query. The addition of a thesaurus can be a very powerful solution for restricted domain applications.

All these techniques are fast, surprisingly accurate and therefore hard to compete with. Another common property is that most of them have been invented over 30 years ago.

The Dilemma of Information Retrieval

Most current IR-systems still use techniques that were developed over thirty years ago and that implement nothing more than a global surface analysis of the textual (layout) properties and simple pattern matching without any understanding of the text or the user at all. No deep structure whatsoever is incorporated in the decision to whether or not retrieve a text.

There is one large dilemma in IR research. The data collections are so incredibly large, that any method other than a global surface analysis would fail. However, such a global analysis could never implement a contextually sensitive method to restrict the number of possible candidates returned by the retrieval system. The study of Information Retrieval has always been much more related to statistical pattern recognition than to symbolic AI techniques.

Information retrieval can also be a very frustrating area of research. Whenever one invents a new model, it is difficult to show that it works better (qualitatively and quantitatively) than any previous model. The addition of new dependencies often results in much too slow a system. Systems such as Salton's SMART exist for over 30 years without having any serious competition [Salton et al., 1968, 1973, 1983a-b,1985, 1987, 1988a-b, 1991], [Salton, 1968, 1971, 1972, 1980a-b, 1981, 1986, 1989].

The field of information retrieval would be greatly indebted to a method that could incorporate more context without slowing down. Since computers are only capable of processing numbers within reasonable time limits, such a method should be based on vectors of numbers rather than on symbol manipulations. This is exactly where the challenge lies: on the one hand keep up the speed, and on the other incorporate more context.

Current Issues

The main objectives of current IR research can be characterised as the search for systems that exhibit adaptive behaviour, interactive behaviour and transparency. More specifically, these models should implement properties for:

- Understanding incomplete queries or making incomplete matches,
- Understanding vague user intentions,
- Ability to generalise over queries as well as over query results,
- Adapting to the needs of an evolving user (model),
- Allowing dynamic relevance feed-back,
- Aid for the user to browse intelligently through the data, and
- Addition of (language) context sensitivity.

In addition to these research topics there is an increased interest in the integrating of traditional databases, free-text systems, multimedia and the addition of (language and common-world) knowledge.

Static and Dynamic Databases

The problem of IR has many facets. The queries as well as the data base elements may be either static or dynamic. Information filtering pertains to static queries in a dynamic data base environment. Here, one teaches a specific interest to a filtering device, which selects interesting text with respect to this interest. Regular free-text search refers to a more static data base with changing queries. Due to its static character, the data base can be preprocessed. In the retrieval phase, one compares the statistic analysis of a query with all the analyses of elements in the data base. Highly correlated analyses suggest a common topic [Croft et al., 1979].

Levels of Analysis

The method of analysis in IR varies between statistical pattern recognition and a symbolic linguistic approach. Clearly, the retrieval quality depends heavily on the amount of context and conceptual knowledge that is available in the retrieval phase. Linguistic approaches result in complicated and computationally complex systems that are not quite relevant in practical implementations. On the other hand, statistical pattern recognition techniques are quite unable to handle conceptual relations and higher order grammatical inferences, which are important to get the retrieval quality above the level of global surface analyses. Generally, IR systems use statistical matching methods on either characters or words and the analysis of meaning does not go beyond the use of synonyms [Van Rijsbergen, 1979], [Lancaster, 1979], [Salton, 1968, 1971, 1980b, 1986, 1989] .

Can one add more (relevant) context to IR by using a Neural Network?

Research in neural networks shows good results in various pattern recognition tasks. Implicit parallelism, easy incorporation of knowledge from different sources, good generalisation and easy association capabilities are the well known examples of advantages of neural networks. So why not use them for yet another classification task: Information Retrieval.

Recent research in connectionist Natural Language Processing (NLP) showed interesting results in self-organising systems [Elman, 1988], [Scholtes, 1991a-i]. Other research shows automatic categorisations of unknown words into clusters which might be used to incorporate a simple notion of meaning in IR [Ritter et al., 1989b, 1990], [Elman, 1988], [Scholtes, 1991a-i]. Although these methods are not capable of analysing complex linguistic structures, they do distinguish different contents better than global surface analyses, while they are still based on automatically derivable training and retrieval algorithms. (And if implemented on parallel hardware, these algorithms would also be fast).

On the one hand, connectionist techniques have the potential to increase the retrieval quality. On the other hand, the IR problem can contribute to the understanding of neural networks as pattern classifiers by comparing neural information retrieval with (already well known) statistical information retrieval results.

2.2 Techniques used in Information Retrieval

There are a number of methods in information retrieval that are difficult to beat.

- Adjacent character statistics or n-gram analyses. A window of size n is shifted over a text. For every possible character combination in a stored document, the frequency is kept. By comparing the frequency vector of a query to that of all documents in the data base, a measure of correlation can be calculated [Forney, 1973] [Hull et al., 1982] [D'Amore et al., 1988] [Kimbrell, 1988]. The main advantages of this method are highly robust behaviour and the elimination of a dictionary. As a result, only a global surface analysis of the text is obtained.

- Inverted indices (representing the exact position of every content word in a stored document) are a well known, accurate and fast technique [Sparck Jones, 1971]. Retrieval with inverted indices can be extended to Boolean queries (A and B) and adjacent queries (A within 5 words of B). The difference between single or multiple occurrences in one document is not measured; therefore, ranking the retrieved documents on basis of their relevance is not possible.

- Next, by giving weight values to the index terms, a so-called relevance ranking algorithm can be designed. Hereby, the relevance value of a document is calculated by multiplying the total occurrence of an index term in a document by a relevance factor. Normally, frequently occurring words have low relevance factors, rarely occurring words have high relevance factors. The documents are ranked in order of their total relevance value. The weights can then be determined manually or automatically. In general, these weights are normalised with respect to the document size [Cooper, 1971].

- Once a certain query is processed, the user can feed the results of a query back into the system. If the user indicates how good or how bad a certain result was, the computer can change the results according to the users input. This technique is called relevance feed-back [Salton, 1989]; it is known to be very effective.

- Instead of using the original word in the index, one can use a subset of artificial (semantic) entities. Each natural word is categorised into one semantical group. The index term only contains the semantical notions: latent semantic indexes [Dumais et al., 1988]. This type of models can be extended to conceptual instead of semantical items, called conceptual information retrieval [Mauldin, 1991]. Both techniques can best be seen as an addition to standard inverted (weighted) indices. Due to its manual nature, the maintenance of semantical groups (or concepts) is expensive. Moreover, semantical groups are subject to personal preferences and once a word is categorised into a certain group, it can no longer be retrieved from others.

- Quite useful is the addition of a (layered) thesaurus to the system in order to provide the user with (semantical) alternatives for key words in his query. The addition of a thesaurus can be a very powerful solution for restricted domain applications.

All these techniques are fast, surprisingly accurate and therefore hard to compete with. Another common property is that most of them have been invented over 30 years ago.

2.3 Evaluating IR: Precision and Recall

Since the beginning of the information retrieval research, the measure of quality has been a severe problem. It is difficult to indicate the exactly result of a query because it is always subject to personal preferences. However, two basic notions are in use: precision and recall.

- Precision indicates the number of correctly retrieved documents relative to the total number of retrieved documents.

- Recall indicates the number of retrieved documents relative to the total number of related documents in the data base.

In general, these two values are said to be inversely proportional, which means that an increase in one of them results in a decrease of the other. Some indicate the performance of their systems by means of these values or a combination thereof. Others compare their system to a well known system. There also are a number of standard test databases which are completely hand-analysed. The main problem here is that these data base are quite small (less than 1 Megabyte, which is absolutely insufficient for proper comparisons of statistical models).

Well known in this context is the study by Blair and Maron. A large number of lawyers were confronted with a legal information system. They were asked to continue searching until they thought 75% of all relevant cases were found. As it turned out afterwards, most of them found no more than 25% [Blair et al., 1985].

In general, these problems are caused by the fact that:

- relevance is a highly subjective concept,

- relevance is always relative to the other documents retrieved, and

- most users don't know what they are looking for.

As a matter of fact, the most interesting queries are the ones in which new answers to new questions are given, a highly difficult task.

3 Neural Computation [2]

"When two elementary brain-processes have been active together or in immediate succession, one of them, on re-occurring, tends to propagate its excitement into the other."

"The amount of activity at any given point in the brain cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportionate (1) to the number of times the excitement of each other point may have accompanied that of the point in question; (2) to the intensities of such excitements; and (3) to the absence of any rival point functionally disconnected with the first point, into which the discharges might be diverted."

-- William James, 1890

In this chapter, neural networks and neural processes are introduced. The discussion covers the first neural models of the forties up to variants of the back-propagation algorithm and the Kohonen feature maps. The aim of this chapter is to take away a little of the magic of artificial neural networks and show that they are no more than well understandable mathematical models for information processing.

Throughout this chapter, many historical anecdotes of the already four decades old neural research direction can be found, so the reader can understand the sensation surrounding the neural research in the proper context.

3.1 Introduction

In recent years, the human brain and neural networks received a tremendous amount of research attention. This chapter focuses on the details of this wave of interest and on general aspects of neural computation. It is organised as follows:

[2] This chapter is partly extracted from "Neural Networks in Natural Language Processing and Information Retrieval", Ph.D. Thesis by Johannes C. Scholtes. Department of Computational Linguistics, Faculty of Arts. University of Amsterdam, January 1993.

First, the problems in symbolic Artificial Intelligence and the appeal of neural networks shall be discussed, explaining why people made the effort to investigate neural networks in the first place.

Computing neural networks are, in many aspects, inspired by their biological counterparts. However, the simplifying assumptions are significant, so one should not take the resemblance very literally. In order to emphasise the distinction we speak of Artificial Neural Networks (ANN's). To avoid hype altogether it might be wise to drop the adjective "neural" altogether and refer to the field as Network Computation. In fact, the general term "neural networks" covers a very large collection of different computing devices. By using a classification scheme that is based on the dimensions: connection, neurone and learning-rule, the reader is provided with an impression of the different types of artificial neural networks.

From the early 40's up to the late 60's a large group of researchers worked with neural networks. As this work is important to understand the value of the current research, the next sub-sections focus on the fundamentals of neurocomputing and on models with imaginative names like Mark-I Perceptron and ADALINE. Best known and largely responsible for the hype surrounding neural networks are two major streams within the neural network research area. First, there is the back-propagation algorithm, as proposed by Rumelhart, Hinton and Williams. Less known, but just as important, is the work done by Teuvo Kohonen on self-organising feature maps at the University of Helsinki. These two algorithms are discussed in the subsequent sub-sections.

Never in the history of computer science has a research topic been surrounded by so many emotionally loaded discussions and debates as the neural network paradigm. These `new' processing elements seemed to conflict with the statistical pattern recognition tradition as well as with the artificial intelligence establishment. Terms like "paradigm shift" and "new definitions of rationality" were used throughout the discussions. Maybe one can understand this sensation better if one has a detailed insight in the rich history of the neural paradigm. Historical facts about neural network research are given throughout this chapter, so one gains a better understanding of this matter.

3.2 Problems in Symbolic Artificial Intelligence

Current symbolic Artificial Intelligence (AI) models suffer from many unresolved problems due to the limitations of the architecture and due to the knowledge representation schemes used. The most important of those problems are:

- To resolve ambiguities, different knowledge sources must be consulted at the same time. In symbolic AI, these sources are normally separated in different system modules. Complex control mechanisms are needed to integrate them properly. Moreover, every system designed to work with separated modules contains communication channels between them. Explicitly defined communication channels between modules show up as bottle-necks in the resulting system.

- In addition, as soon as some input value is slightly corrupted or disturbed by environmental noise, the entire system collapses. This inability to reason with incomplete or imprecise information is caused by the local data representation schemes that can be found in symbolic AI. These systems use memory models where one concept is stored in one particular slot. As soon as the slot is damaged or the input element changes a little, it can no longer be found in the memory bank, resulting in a system failure. Due to this strictly local data representation, generalisation and error-correction are impossible.

- Most important and so far not explained by symbolic AI is the fact that humans have an adaptive way of learning. In symbolic AI, one must pre-code and pre-categorise real-world data into groups of symbols, before one actually starts to define relations between them. This results in problems in the long run. A system that is capable of discovering categories by itself would be preferred.

- Symbolic AI systems tend to be slow. On the one hand this is caused by the inability of current computer systems to process symbolic information. Vector representations can be processed much better. On the other hand, it is virtually impossible to parallelise symbolic AI techniques due to the strongly sequential nature of these techniques.

The next sections will discuss these problems in more detail as they occur in symbolic information retrieval (IR). At the end, a motivation for the use of connectionist methods shall be given.

Ambiguity

In spite of the fact that ambiguity has always been one of the most important problems in natural-language processing and natural-language understanding, it was rarely solved by symbolic AI. One can understand that determining the correct meaning of a word is one of the basic functions of an IR system. In a way, the philosophy behind sequential-symbolic IR-system architectures caused the lack of an efficient disambiguation model.

Different types of lexical ambiguity can be distinguished. One type is syntactic-lexical ambiguity: correct categorisation of a word, e.g., noun verses verb. There are two types of semantic ambiguity. The first one is called polysemy: words with several meanings that are related ("The government fell" versus "John fell and hurt himself"). The second, homonymy, pertains to unrelated words with the same form ("foot-ball" versus "dance-ball"). To resolve these types of ambiguity, IR systems must take various sources of knowledge into account.

Because most IR systems work sequentially (first syntactical, then semantical analysis), it is very difficult to solve the ambiguity problem. One cannot resolve the complete ambiguity in the syntactical part without taking into consideration the semantical disambiguation.

The same argument holds for the semantical phase. If ambiguity is not resolved in the syntactical phase (i.e. all the various meanings are passed to the semantical phase), the semantical disambiguation definitely becomes too complex. There is a need for a method that solves the ambiguity problem by taking into account all the available knowledge at once.

Blackboards [Hayes-Roth, 1985], backtracking [Woods, 1970], delay [Marcus, 1980] and marker-passing [Charniak, 1983] are efforts to solve the problem by integrating different knowledge sources. However, the bandwidth between the different sources seemed to be broader than expected. Combining syntactical and semantical knowledge in one distributed knowledge base might do better.

Robustness

Humans often use language incorrectly; still one understands what someone else means if one communicates with him or her. Errors are made at different levels such as spelling, grammar and meaning. Because current IR systems do not work with content-addressable memory and association (at all levels) it is very difficult for symbolic methods to recognise and correct an incorrectly used word. The correction of a misspelled word in a non-associative memory system is an $O(n^3)$ problem, where n is the number of possible words [Wagner et al., 1974]. This complexity can be brought back to $O(n^2)$ by using extra memory, but it remains complex. Solving a syntactical or semantical error is even worse.

Because of their architectural shortcomings, sequential IR systems with classical memory utilisation schemes are unable to solve these problems in an elegant way.. Whenever an IR system fails to work because one word has been entered wrongly into the system, this might be called a basic shortcoming.

Learning and Generalisation

One of the disadvantages of symbolic IR systems is their inability to learn automatically. Everything has to be encoded by hand: lexicon, syntax rules and semantical issues. Once the system encounters an unknown word combination, grammatical use or meaning, it does not know what is meant. Several researchers tried to solve this problem. However, symbolic methods seemed to be too complex for efficient learning algorithms [Kodratoff et al., 1990], [Michalski et al., 1986a, 1986b] [Winston, 1975].

Related to learning is the aspect of generalisation. Once a system has certain information, it might be able to derive the meaning of unknown language use by generalising. To accomplish this, high level integration of all the system's knowledge is needed, since the architectural limitations of symbolic-sequential IR systems do not meet this demand. This might be the reason why these systems still lack the ability to generalise. The term learning can also be positioned in a more general cognitive context: the problem of language acquisition.

The symbolic AI community never quite tackled the problem of language acquisition, probably the most interdisciplinary study in cognitive science.

Complexity

The complexity of natural language has forced researchers to split the problem into sub-problems. The nature of sequential computers made them decide to use sequences of modules which interact at different levels. However, the interaction between the different partial solutions was too intense, so the systems were suffering from communication bottle-necks (physically as well as conceptually).

The hope that future computers would be more powerful gave developers of the computationally complex solutions an excuse and hope to proceed. Although computers became more powerful than anybody expected, the complexity problem is still not solved. Therefore a theory of language processing is needed that, instead of simple passing of semi-complete results between processing parts, posits strong (parallel) interaction between those components.

Much used in symbolic IR systems are rule-based knowledge representation methods. In the eighties these methods were mainly used because of their flexibility and expressive power. It turned out that the flexibility led to rule bases that were difficult to maintain, in particular when they were big. Moreover, it was difficult to split certain problems into separate sub-problems (needed for better maintenance, reduction of complexity and easier parallelisation). Many times, rule based systems resulted in spaghetti-bases.

The Motivation for Neural Networks in IR

If symbolic AI systems still suffer from such fundamental problems (after so many years of research) one may question their ability to model problems involving these forms of intelligent behaviour. It may be useful to study other computing models such as artificial neural networks, because:

- A distributed representation might increase the system's ability to integrate different sources of knowledge in an elegant and effective way. By seriously integrating different kinds of knowledge, perception and common sense have a better chance of being achieved.

- Neural networks can correct corrupted real-world input. An even more interesting aspect of neural networks is the ability to generalise over known knowledge. In this case, the system can react properly to unknown input.

- Directly resulting from the architecture is the computational power of a neurally inspired system. Natural language understanding has shown to be a NP-complete problem.

> Connectionist models provide a solution to the computational complexity by enabling highly parallel computations.

Presumably most important is the argument that people do not have a store of knowledge (i.e. tables, addresses and pointers), people are knowledge. Neural networks provide a framework to implement such a memory-based model.

3.3 Neural Network Models

Background

In general, the last fifty years of research in the field of machine intelligence can be characterised by the following global periods. In the 1940s people started to explore simple neural networks. These simple networks were all constructed by hand. There was no notion of learning whatsoever. In the 1950s, the focus was on learning. By automatically adapting the weights between the neurones, long term memory could be implemented in the machines. The beginning of the 1960s was dominated by the connectionist approach. Most important were mathematical and biological influences from cybernetics and neuro-psychology. At the end of the 1960s the symbolic approach became increasingly popular. In the 1970s, focus was on the representation of knowledge; expert systems and knowledge based systems. The 1980s were under the spell of the revival of (neural) learning machines. As far as the 1990s have proceeded, a trend towards biological neural networks and statistical learning algorithms can be noticed.

The first people to implement a neural network in hardware were Marvin Minsky and Dean Edmonds in the summer of 1951. The memory of this machine (called the SNARK) was stored in the position of the control knobs. The machine was made of 40 of such knobs. When the machine was learning, it changed the position of the knobs, using a B24 bomber gyropilot. Minsky was so amazed by the functioning of this machine, that it stimulated him to write his Ph.D. thesis on a problem related to machine learning. This machine was so complex, that they could never debug it 100%. However, "its random design was almost sure to work no matter how you built it" [Bernstein, 1981].

The Basic Elements of an Artificial Neural Network

Four decades of research in neural networks provide us with more than fifty different neural network structures. Overall, it can be stated that a neural network is characterised by three main features: the distributed processing elements, the connections between them and the learning rule. Although all neural networks have this in common, lately many variations have been invented.

In [Lippmann, 1987] a good overview of the most popular algorithms and models of this confusing field is given in a clear and intelligible way.

Data Representation

In localist connectionism, each concept or hypothesis about the environment is represented by one unit. Relations are represented by two types of links: inhibitory and excitatory, both having the same functionality as synapses in biological neural networks. Most localist networks show a hierarchical structure. The disadvantage of local representations is the fact that it is quite hard to implement connectionist properties such as robustness and adaptive behaviour. Therefore powerful heuristics and techniques need to be developed. There is a parallel between this approach and symbolic artificial intelligence, with all the problems discussed earlier.

Distributed connectionism on the other hand, is based on representing concepts as patterns over large numbers of units. Each unit represents a micro-feature of the pattern. Similar concepts share similar micro-features and therefore have similar patterns. Distributed connectionism provides an efficient way to represent knowledge. Moreover, it provides us with a self-generalising, associative representation. Mainly because of this last reason, distributed representations must be preferred over local ones [Feldman et al., 1982].

Activation Functions

Apart from the general structure of the neurone, an activation function must be defined, meaning, the exact manner a neurone reacts to input activations. Biological activation functions have very non-linear characteristics. Artificial activation functions are very simple abstractions of such functions. The activation (firing rate) of a neurone is a function of the network activity. Each firing event is called a *spike*. In mathematical terms, an activation x_i of neurone i as a function of the activity of the network elements that are directly connected to neurone i: net_i, can be written as:

$$x_i = f(net_i) \qquad \text{(EQ 1)}$$

where w_{ij} is the weight of the connection between neurone i and j, x_j is the output activation of neurone j, and net_i is the activity received by neurone i from the entire network.

In general, three different types can be distinguished: the linear, the threshold, and the sigmoid activation functions. First the linear:

$$f(net_i) = \sum_j w_{ij} x_j \qquad \text{(EQ 2)}$$

The linear activation function is the simplest one. In this case:

$$x_i = f(net_i) = c \cdot net_i \qquad \text{(EQ 3)}$$

where c is a constant, also called the gain of the activation function (see "Linear activation function"). c is the same for the entire network.

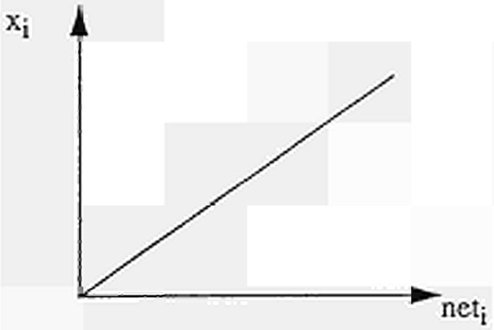


FIGURE 3.1: LINEAR ACTIVATION FUNCTION

A better approximation of a biological activation function is a (relay) threshold function. In this case, the neurone only fires at a certain input activation.

$$x_i = f(net_i) = \begin{cases} 0, if\,(net_i < a) \\ 1, if\,(net_i \geq a) \end{cases} \qquad \text{(EQ 4)}$$

where a is the threshold value (see "Threshold activation function"). This value is the same for the entire network.

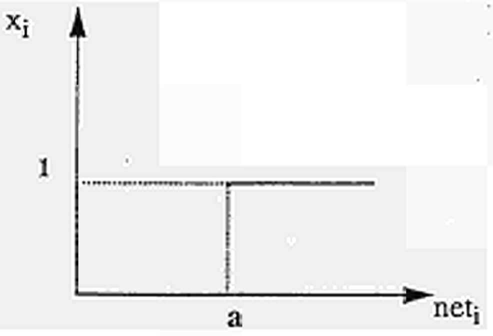


FIGURE 3.2: THRESHOLD ACTIVATION FUNCTION

This function is non-linear and non-continuous, which causes great difficulties in mathematically analysing the behaviour of the model. Therefore, a variation can be found in the literature, the semi-linear threshold function. This function has a linear behaviour between a minimal and a maximal value of the input activation. This function has the advantage that it approximates the biological activation functions well enough, while keeping a linear character. A disadvantage of this function is the fact that there are discontinuities in the first derivative.

$$x_i = f(net_i) = \begin{cases} 0, (net_i < a) \\ c \cdot net_i, (a \le net_i \le b) \\ 1, (net_i > b) \end{cases} \qquad \text{(EQ 5)}$$

where a is a lower threshold, b an upper threshold and c a constant equivalent to the gain in the linear activation function. Sometimes this function is referred to as a two-level threshold function (see "Semi-linear threshold function").

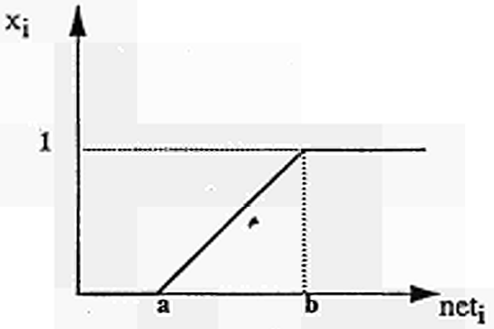


FIGURE 3.3: SEMI-LINEAR THRESHOLD FUNCTION

The best non-linear function which is continuous in all derivatives, is the sigmoid function. This is a non-negative, everywhere monotonically increasing function that approaches zero and one respectively for $x \to \infty$ and $x \to -\infty$. The sigmoid function approximates the biological activation functions quite well.

$$x_i = f(net_i) = \frac{1}{1+e^{-(\lambda \bullet net_i)}} \qquad \text{(EQ 6)}$$

where λ is also called the steepness of the activation function. The problem of this function is the non-linear character. As a result, it could not be applied until a training method was developed that could handle non-linear activation functions (see "Sigmoid threshold function").

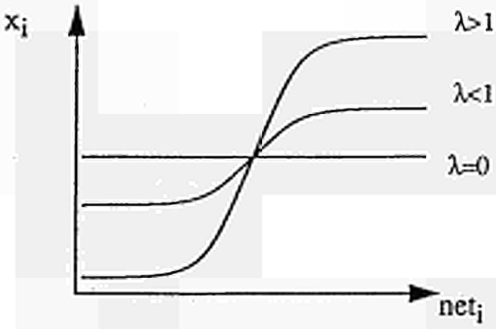


FIGURE 3.4: SIGMOID THRESHOLD FUNCTION

The McCulloch & Pitts model uses a threshold activation function. The ADALINE and Perceptron model use linear activation functions. The non-linear sigmoid activation function is often applied in back-propagation neural networks.

Network Topology

Two main-stream network topologies can be distinguished: the feed-forward and the recurrent (or feed-back) neural networks.

- In feed-forward networks, the output activation is in one direction only: forward. The neural network has input activity on one side, then perhaps a number of in-between layers and then an output activity on the other side. There are no recurrent connections. Good examples of feed-forward networks are the Perceptron, ADALINE and the back-propagation network (also called multi-layer perceptron).

- Recurrent networks do have feed-back connections. This means that the output activation of the network at a certain time depends on the input, as well as on the output activation of the neural network. Therefore, recurrent neural networks have very complex dynamical properties. Some examples of recurrent neural networks are associative memories, the Hopfield model, Kohonen feature maps, ART (1 and 2) and the Simple Recurrent Network (SRN).

The McCulloch & Pitts model actually only defines a neurone. Because these neurones can be connected in any possible way, the McCulloch & Pitts model does not have a place in this classification.

Learning Rules

Two different kinds of learning can be distinguished: supervised and unsupervised learning.

- In a supervised learning model, there is an explicit external teacher that provides the network with input-output pairs. The weights of the network are adapted by a function of the desired and the current output value for a specific input activation.
- Unsupervised models adapt the connection weights by taking into account the activations of the neighbouring (clusters of) neurones. There is no external teacher and there is no pre-defined set of categories of which the input stimuli are part. The term self-organisation is often used in relation with unsupervised models.

The supervised as well as the unsupervised learning rules are all derived from the famous Hebb learning rule, described by Donald Hebb in his 1949 work on synaptic learning: "The Organization of Behaviour". Hebb did not include any quantitative descriptions of synaptic adaptation, but was close enough when he stated:

> *"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased".*

The Hebb rule states that if two neurones *i* and *j* are activated simultaneously, the strength of the connection between them w_{ij} (weight) must be increased.

$$\Delta w_{ij} = \varepsilon \cdot x_i \cdot x_j \qquad \text{(EQ 7)}$$

where ε is the so-called learning rate and x_i and x_j are the activation values [Hebb, 1949].

Besides this very important rule, Hebb also proposed the idea of short term memory as a recurrent connection between (or within) neurones and the idea of distributed representations.

A real Hebbian way of learning can be found in [Malsburg, 1973] and [Linsker, 1988]. In these models, there are no restrictions to the interconnections of the model. All interconnections can be learned, the inter-layer as well as the one-layer inter-neuronal connections. Firing may be as non-linear as one wishes it to be, i.e. sigmoid, threshold or delay functions can be used. Although this model is biologically plausible and its results are promising, it requires a lot of computation and is therefore not popular at all within the already complex NLP applications.

The perceptron, ADALINE, and the back-propagation neural network (supervised learning models), as well as associative memories, Kohonen feature maps, the Simple Recurrent Network (SRN), and ART (unsupervised learning models) use Hebbian learning rules.

The McCulloch & Pitts Neurone

Some say the neural network paradigm started with the famous McCulloch & Pitts paper in 1943. Others say the start goes back to the work of the psychologist William James in the 1890s. Whatever is right, neurone-like computer architectures have inspired researchers for many years. James' work suggested many of the neural network paradigms as they are known today (although he was not aware of the precise operation of these computational elements). In the late 30s and early 40s, other researchers like Norbert Wiener and even John Von Neumann (whose name is given to modern serial computer architectures with their so-called Von Neumann bottleneck) suggested brain-like research. The real breakthrough had to wait until the middle of the second world war.

In 1943, Warren McCulloch and Walter Pitts wrote "A Logical Calculus of the Ideas Immanent in Nervous Activity". In this paper they proved that the basic elements of the logical calculus: AND, OR and NOT could also be implemented in small, neural-like computing elements: the McCulloch & Pitts neurone. So, any finite logical expression could be realised by a McCulloch & Pitts network.

Each neurone has a fixed threshold. Depending on the input it receives through excitatory and inhibitory synapses, it fires a signal through the axon if the total input exceeds the threshold of the neurone. Although the networks had to be constructed by hand (learning was not possible), this work showed the possibilities of small computing neural networks in detail. Moreover, it had an immense influence on neuroscientists as well as on computer scientists [McCulloch et al., 1943].

A neurone was modelled as a logical threshold unit. A neurone has a number of binary input signals x, and one output signal y. Neurones can be linked together, so the output of one neurone is the input for the next. The output activation y is determined by a threshold function that fires if the addition of the input activities from neurone i, x_i, times the weights of the connection with neurone i, w_i, exceeds a certain threshold s.

$$y = \Theta(\sum_i w_i x_i - s) \qquad \text{(EQ 8)}$$

where Θ is a threshold function of the form:

$$\Theta(x_i) = \begin{cases} 0, (x_i \leq 0) \\ 1, (x_i > 0) \end{cases} \quad \text{(EQ 9)}$$

It was shown by McCulloch and Pitts that any arbitrary logical function (such as AND, OR, NOT, XOR) could be constructed by a combination of such elements (see "The McCulloch & Pitts logical threshold unit").

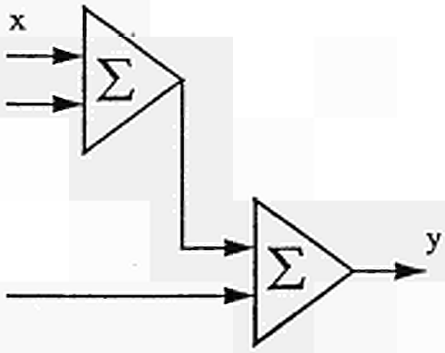


FIGURE 3.5: THE MCCULLOCH & PITTS LOGICAL THRESHOLD UNIT

Although McCulloch and Pitts started the neural networks research, they dealt with logical circuits rather than with neural networks in the current sense. Two important aspects they did not take into account are:

- Training of the weights. All weights in their models were hand-constructed.
- Biological neural networks show significant redundancy. This means that a number of neurones can be eliminated without influencing the functionality of the model too much. In the Logical Threshold Unit, one defect neurone directly influences the output activations. This problem is caused by the local data representation of the model.

The Perceptron

Frank Rosenblatt, a Bronx High School of Science classmate of Minsky, introduced a new phase in neural research with his 1958 paper on "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain". His invention (the "Mark I Perceptron") was important in more than on way. First, this was finally a computing device that *did* something. Originally Rosenblatt was a psychologist, and so his perceptron did what a psychologist thought was important. Second, this was a machine that was capable of learning something, and that was what engineers wanted to put their hands on. Finally, although simple at first sight, perceptrons were mathematically extremely complex which made them interesting for mathematicians studying complex non-linear systems.

A perceptron is a supervised feed-forward network that uses binary (linear) neurones. Perceptrons performed remarkably well in pattern classification. In fact, perceptrons were taught to give transformations from an n-dimensional feature $\{0,1\}^n$ space to $\{0,1\}$.

Rosenblatt used his perceptrons mainly to classify grey levels of image pixels into characters or shapes. This was primarily caused by the fact that he modelled the perceptrons according to biological nervous system structures he found in the retina [Rosenblatt, 1958].

Learning was implemented by a simple *reinforcement rule*. This rule could be self-organising as well as supervised (or *forced adaptation* as Rosenblatt called it). The perceptron model consists of two layers: the *A* Units (on which the object is projected), and the *R* Units which classify the object. The *R* Units are randomly connected to the *A* Units. The weights between the *A* and the *R* Units can be changed. If an *R* Unit is activated at a certain moment in time and a connected *A* Unit is also activated, then the connection strength between them is increased.

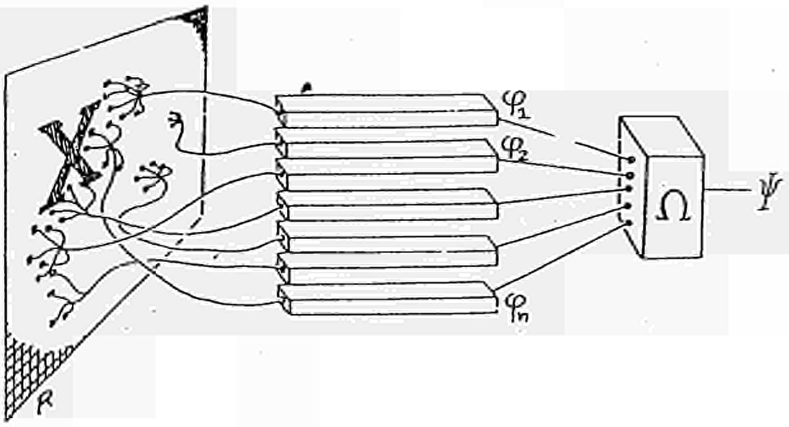


FIGURE 3.6: THE PERCEPTRON (REPRINTED FROM [MINSKY ET AL., 1969/1988]).

Perceptrons use the following activation function:

$$y_i = \Theta(\sum_{i=1}^{n} w_{ij} x_i) \qquad \text{(EQ 10)}$$

where Θ is a so-called step function.

In 1962, Rosenblatt introduced his Perceptron convergence theory, in which he explained and proved the correctness of his perceptron learning rule [Rosenblatt, 1962]. The learning algorithm works as follows:

1. Provide all connections with random weights.

2. Select an input vector from the training set.

3. Feed-forward this vector. If the perceptron responds properly, do nothing. Otherwise, change the connections.

4. Continue with step 2 until the network classifies the training data correctly.

The most important result of this theory is the *Perceptron Convergence Theorem*, which states that:

> *"If there exists a set of connection weights which is able to classify the input patterns in the corresponding classes, then the perceptron learning rule will converge to this set of weights in a finite number of steps, regardless of the initial set of weights."*

The Mark I Perceptron was the first successful neurocomputer. It was trained to recognise characters from a 400 pixel image sensor. The Mark I Perceptron could store up to 512 weights. This computer appeared to be very successful in its task and the world expected (too) much from this machine. As Rosenblatt stated:

> *"The question may well be raised at this point of where the perceptron's capabilities actually stop....the system described is sufficient for pattern recognition, associative learning, and such cognitive sets as are necessary for selective attention and selective recall. The system seems to be potentially capable of temporal pattern recognition....with proper reinforcement it will be capable of trial and error learning, and can learn to emit ordered sequences of responses."*

Although another 30 years passed before some of these aspirations of perceptrons were fulfilled, they were not too ambitious.

To appreciate exactly how enthusiastic Rosenblatt was consider this phrase from [Rosenblatt, 1958]:

> *"It seems clear that the class C perceptron introduces a new kind of information processing automaton: For the first time, we are having a machine which is capable of having original ideas. As an analogy of the biological brain, the perceptron, more precisely, the theory of statistical separability, seems to come closer to meeting the requirements of a functional explanation of the nervous system than any system previously proposed."*

Less well known is the fact that Rosenblatt himself was also aware of the limitations of his computing devices; in particular in the case of relative judgements and symbolic behaviour. In these cases, Rosenblatt described the behaviour of his perceptrons as that of "brain damaged patients" [Anderson et al., 1988] [Hecht-Nielsen, 1990].

The ADALINE

The Perceptron Convergence Theory, as proposed by Rosenblatt, sometimes resulted in very long training times, and if finally a set of weights was found, it was not known whether it was the most optimal one.

In 1960, Bernard Widrow and Ted Hoff proposed a faster type of neurone learning algorithm. The computing device belonged to the family of perceptrons. It was called an "Adaptive Neuron" and was implemented by a Threshold Logic Unit with variable connection strength. Applied to the ADALINE, Widrow and Hoff were able to develop a supervised training algorithm for single layer neural networks: *the delta rule.*

Almost all of today's learning rules are derived from the delta rule. For instance, the backpropagation algorithm is a generalised case of the delta rule for multi-layer perceptrons with non-linear activation functions.

The ADALINE was the first learning model for continuous signals. One of the main advantages of the ADALINE was that it could easily be implemented in hardware. The model uses linear activation functions:

$$y_i = \sum_{i=1}^{n} w_{ij} x_i + w_0 \qquad \text{(EQ 11)}$$

The problem with the Hebb rule is that the value of the weight continues to increase as the neurones *i* and *j* are activated. The delta rule suffers no longer from this problem. However, as a result, the delta rule is a supervised training rule, where the original Hebb rule is unsupervised. The delta rule follows the equation:

$$\Delta w_{ij} = \varepsilon(y_i^{(corr)} - x_i)x_j \qquad \text{(EQ 12)}$$

where ε is the learning rate, $y_i^{(corr)}$ is the desired activation for neurone i and x_i is the actual activation of neurone i.

Because the ADALINE could easily be generalised towards larger networks, Widrow and Hoff tried to develop a multi-layer supervised training algorithm for many years. Unfortunately, they did not succeed. The world had to wait for the back-propagation algorithm to implement such a functionality. Nowadays, everybody who owns a facsimile machine or a 2400 baud modem also owns an electronic implementation of a Widrow and Hoff "Adaptive Neuron" [Widrow et al., 1960].

The XOR Problem

By the end of the 1960s, the first golden period of neural network research ended with the book "Perceptrons, An Introduction to Computational Geometry" by Marvin Minsky and Seymour Papert. In this very important work they showed that the single layer perceptron could never learn the Exclusive-OR problem. In fact they proved that there is no (single layer perceptron) weight set for the XOR problem by three simple graphs (which were proceeded by an extensive geometric analysis of the behaviour of perceptrons).

Assume there is a single layer perceptron with two inputs. Then, the activity of the network can be calculated as:

$$y = w_1x_1 + w_2x_2 + w_0 \qquad \text{(EQ 13)}$$

For the AND, OR and XOR function, the input space can be represented geometrically in the following graphs. Open points indicate a zero as output of the perceptron, closed points indicate a one as desired output.

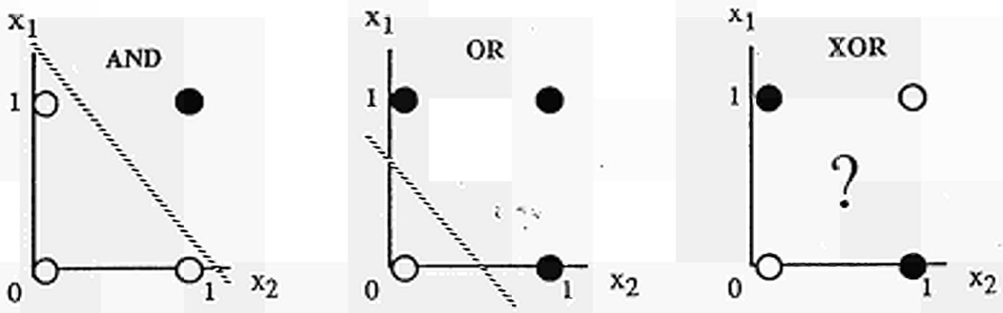


FIGURE 3.7: GEOMETRICAL REPRESENTATION OF THE PERCEPTRON'S INPUT SPACE

Four points exist: (0,0), (0,1), (1,0) and (1,1). It may be clear that in the XOR case, one cannot use a linear classifier to separate the open from the closed points. Therefore, the (single layer) perceptron cannot learn the XOR problem [Minsky et al., 1969/1988].

By introducing a hidden layer one can extend the perceptron in such a way, that in principle it can make the classifications necessary for the XOR problem. In those days however, one did not know how to train a multi-layer perceptron, so this was not a relevant solution. In a multi-layer model, it is not known which neurone on which layer is responsible for a certain output value. Therefore, one does not known which weights to adapt. This problem is known as the *credit-assignment problem*.

One important issue was overlooked in this period. Perceptrons might not have been able to learn some mathematical functions in finite time, but they were very well able to model psychological behaviour. For most researchers that should have been much more important than the message from Minsky and Papert. Nevertheless, this message did cast serious doubts on the viability of neural networks.

To some extent, the decline of the neural research school may have been due to "Lord" DARPA (Defense Advanced Research Project Agency) who decided not to spend any more money on a research direction that could not even solve the problem of exclusive OR. Much more was expected from the other sister of Machine Intelligence: the symbolic Artificial Intelligence community. Due to lack of funding, the neural paradigm went underground for more than a decade.

The Dark Ages

Researchers like Stephen Grossberg, Leon Cooper (a Nobel laureate in superconductivity), Christopher Von der Malsburg, Shun-Ichi Amari, Kunihiko Fukushima, David Marr, Teuvo Kohonen and James Anderson continued working on neural networks during the 70s. The last two are particularly well known for their simultaneous independent publications on associative memory models in 1972. These memory models used linear neurones and a Hebb-like training rule [Anderson, 1972][Kohonen, 1972, 1977, 1984].

Stephen Grossberg produced an enormous amount of research papers and several books from the beginning of the 70s up to now (most of which were very mathematical and difficult to understand on their own). Only recently did he start to simulate his models. Together with his group (and his wife, Gail Carpenter, in particular), he developed the ART and ART-2 implementations and used them in various applications. ART-2 was the first patented neural network algorithm.

By the beginning of the 80s there were a number of different influences that triggered the revival of the learning machine. First, James McClelland and David Rumelhart developed a psychologically plausible model of character and word recognition. Although their model did not implement any learning (the weights of the connections had to be set by hand), it showed that characters in words were better and faster recognised than the same characters in non-words by implementing an efficient "Interactive Activation" model [McClelland et al., 1981][Rumelhart et al., 1982].

The Renaissance

In 1981 Richard Sutton and Andrew Barto analysed the Delta Rule in much detail, providing the community with a much better understanding of this basic mechanism [Sutton et al., 1981].

The modern era of neural networks really starts with the publication of John Hopfield's 1982 paper on "Neural Networks and Physical Systems with Emergent Collective Computational Abilities". John Hopfield was known as a distinguished physicist. His decision to study neural networks was very important for the field to be taken seriously. In fact, by relating the training of a network to seeking minima in energy landscapes he laid a theoretical foundation for neural research [Hopfield, 1982].

From that moment on, it all seemed to happen simultaneously. Some important milestones achieved in those days were:

- Jerome Feldman and Dana Ballard from Rochester University, who made the term Connectionism popular, pointed out the biological implausibility of many of the Artificial Intelligence approaches in [Feldman et al., 1982].

- David Ackley, Geoffrey Hinton and Terrence Sejnowski developed the "Boltzmann Machine", a neural model similar to Hopfield's model but extended with a stochastic element. Moreover, they introduced "stimulated annealing", a technique to prevent a model from getting stuck in local minima.

Back-propagation

The real boom came after the 1986 publication of David Rumelhart, Geoffrey Hinton and Ronald Williams "Learning Internal Representations by Error Propagation." By showing the world how to train a multi-layer perceptron model with non-linear neural activation functions, a whole new research field appeared. Just as Teuvo Kohonen and James Anderson simultaneously published their work on associative memories, so was the back-propagation algorithm invented by different people at the same time. The credit for the algorithm goes to [Rumelhart et al., 1986a], who gave it the name "error back propagation" and related it to neural network research.

The two discoveries of the algorithm simultaneous to that of Rumelhart et al. in the mid 80s were by Yan Le Cun and David Parker [Le Cun, 1986][Parker, 1985]. However, in 1974, Paul Werbos already described an algorithm similar to back-propagation in his Harvard Ph.D. thesis [Werbos, 1974]. Some even go further back in history to seek the real roots of the back-propagation algorithm. Similar mathematical recursive control structures can be found in the work of Arthur Bryson and Yu-Chi Ho in 1969 [Bryson et al., 1969]. It can even be shown that the structure of the learning algorithm follows from the so-called Robbins-Monro technique as introduced in 1951 [Robbins et al., 1951].

As mentioned, the back-propagation algorithm became the backbone of neural research. Every day, new applications appeared on the electronic news groups. The first IEEE neural network conference in 1987 in San Diego received over a 1,000 research papers. Particularly famous became the NETtalk implementation. So far, nobody had been able to develop a model that could learn how to pronounce English words. NETtalk could. Now, identical implementations are known for German and Dutch [Sejnowski et al., 1986][Dorffner, 1988][Weijters, 1990].

The back-propagation algorithm is a supervised learning algorithm. A back-propagation network consists of multiple layers: an input layer, one or more hidden layers (usually only one), and one output layer. By taking less neurones in the hidden layer than in the input and output layer, a distributed data representation in the hidden layers is forced, which yields the generalisation and association behaviour of the model. The errors that are used by the learning rule for the adaptation of the weights of the hidden units are back-propagated from the errors found in the output units (see "The back-propagation network model").

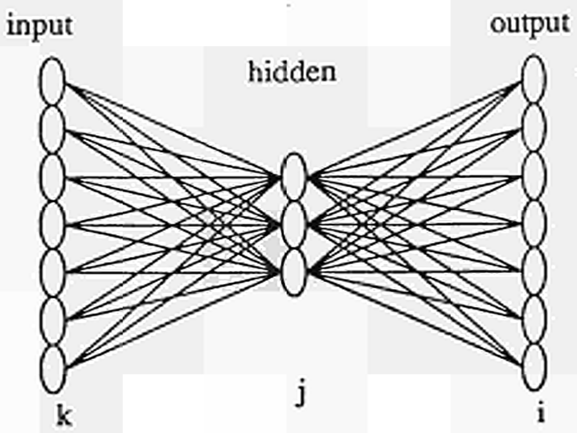


FIGURE 3.8: THE BACK-PROPAGATION NETWORK MODEL

If *i*, *j* and *k* represent the neurones in the output, hidden and input layer, then the activity s_j of neurone *j* of the hidden layer is given by:

$$s_j = \sigma(\sum_k w_{kj} s_k) \qquad \text{(EQ 14)}$$

where σ is the so-called sigmoid function. See "Activation Functions". The activity functions of the output layer are:

$$s_i = \sigma(\sum_j w_{ij} s_j) \qquad \text{(EQ 15)}$$

The output error for the v^{th} pattern in the i^{th} output neurone can be defined as:

$$\varepsilon_i^v = y_i^v - s_i(x^v) \qquad \text{(EQ 16)}$$

A sigmoid function is used for which holds:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \qquad \text{(EQ 17)}$$

Then the following weight update rules for the back-propagation rule can be given:

$$\Delta w_{ij} = \alpha \cdot \varepsilon_i^v \cdot s_j s_i (1 - s_i) \qquad \text{(EQ 18)}$$

$$\Delta w_{kj} = \alpha \cdot \sum_{i} \varepsilon_i^{v} \cdot s_k s_i (1 - s_i) \cdot w_{ji} \cdot s_j (1 - s_j) \qquad \text{(EQ 19)}$$

where α is a small constant.

Although the back-propagation algorithm meant a great step forwards for neural networks as a computing paradigm, there are a number of basic problems.

- First, the error function is a very complex function of all the connection weights. This function has numerous local minima. The gradient descent always leads to the nearest minimum, which may be higher the any global minimum.
- Next, the back-propagation algorithm has a number of learning parameters, which are not given by the algorithm. The convergence of the back-propagation algorithm depends greatly on these initial values. So, wrong initial values can cause wrong weights and thus a wrong mapping. Determining the proper learning parameters is often based more on guessing than on good reason.
- Third, the algorithm can only be trained with pre-obtained structured learning sets, meaning, combined input-output sets must be acquired. In many cases this is not possible as a result of: insufficient knowledge of the system behaviour (e.g., what are the required output sets), insufficient number of measurements, and errors as a result of wrong manually structuring of the input-output sets.
- Finally, it may occur that the network has a very high (or very low) activation value, resulting in an activation value (due to the sigmoid function) near one (or zero). Then the weight adaptation shall be minimal and as a result, the training process will come to a complete standstill [Ritter et al., 1992].

Associative Memories

One of the first written works on memory can be found in [Aristotle, 400 BC]. Aristotle described the basic properties of what is known today as an associative memory. "We do not have an explicit store of knowledge, we are knowledge" was one of the statements in the beginning of this chapter. Aristotle was one of the first who was aware of this. He described a memory system in which objects are stored within a certain distance of related objects given certain *features*. Today, we call such memories: *associative memories*.

In particular, Aristotle defined the basic element of memory as a sense image. In addition, he defined associations as links between these memories. These links then served as the basis for higher-level cognition. Although Aristotle was quite unaware of the exact functionality of the memory elements, he tried to solve some typical memory problems. For instance the known problem of different versions between sense images in time was solved by tagging the various versions with temporal information. Finally, he proposed a method to compute with these images by using the links. Different types of links can cause association: temporal links, "something similar" links, "opposite" links, "neighbouring" links. Aristotle defined this association process as a highly dynamical and flexible process, even as a kind of searching.

Associative memories have many important properties such as fault-tolerant processing, implicit generalisation, and automatic error-correction. But, what is meant by distance and what are the exact features to base the distance on? Moreover, how are these features derived, expressed or normalised? And how can the feature values of these associative memories be determined automatically?

Models such as the perceptron and back-propagation are interesting (supervised) pattern classifiers. They can be trained to represent (non-linear) function mappings. But they are not capable of implementing an associative memory in which related object are in neighbouring areas. However, Kohonen feature maps, ART and some of the other self-organising models can implement such a process.

In cognitive processes (of which language is one), associations and generalisations can be modelled much easier by using such an associative mechanism. One just has to check the direct neighbourhood of a particular concept in order to find related concepts. The really interesting aspects of (high-level) cognitive behaviour can probably be modelled better with self-organising models instead of with function approximators such as back-propagation.

The Simple Recurrent Network (SRN)

In [Elman, 1988], the author uses a recurrent network, feeding back the hidden layer units into the input layer. By giving the first and second element in a string, the network can predict the third one. Elman uses this capability to learn the network simple sentences. After learning it was capable to determine the grammatical correctness of a sentence. The system could even categorise words into syntactical groups, without any notion of grammar.

The automatic determination of linguistic results obtained by Elman has been analysed by many other researchers. [Servan-Schreiber et al., 1988, 1989, 1991] and [Cleeremans et al., 1989, 1990] showed that these networks can determine the grammatical correctness of finite state grammars. Finite state grammars are among the simplest grammars: the appearance of a symbol in a string is determined by the precedents in that string. However, natural language is known to be right-side (strings to come in the near future) dependent too.

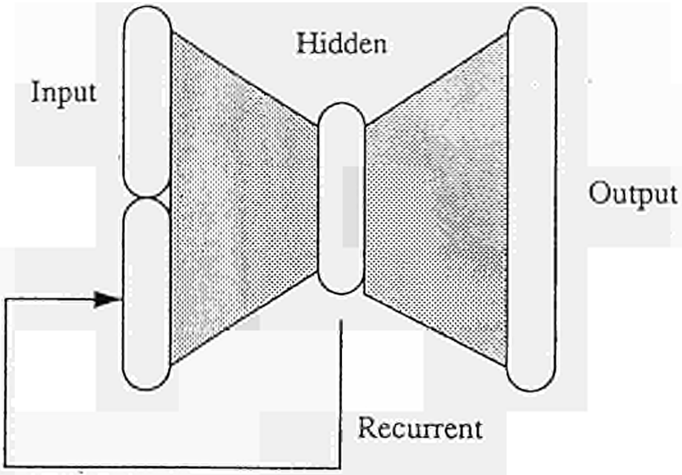


FIGURE 3.9: THE SIMPLE RECURRENT NETWORK (SRN): FEED-BACK OF THE HIDDEN UNITS INTO THE INPUT UNITS.

In [Allen, 1990] and [Stolcke, 1990] the authors demonstrate even more properties and functional capabilities of networks with comparable architectures, learning rules and computational power. Although these networks embody the usability of recurrent neural networks for serial processing, finite-state grammars are insufficient for realistic natural language processing.

It is proven that Elman's model is more powerful than Jordan's model in [Cottrell et al., 1989]. Here the authors show that there are certain tasks such as counting that cannot be taught to a Jordan-style network whereas a Elman-style network can learn this behaviour up to a certain limit.

However, the SRN reveals unstable behaviour if the training set or the neural network becomes too large. Elman introduced a method which he called incremental learning [Elman, 1991b]. Here he suggests one should start with a small training set that slowly grows toward a more realistic representation of the real problem.

Somehow, Elman has not completely solved the problems occurring in his model. First, the SRN is often criticised because it ignores structure completely. Next, large SRNs still have instable characteristics despite attempts from Elman to prove otherwise. Finally, the SRN cannot handle recursive structures, resulting in an explosive growth of the number of needed neurones when one tries to implement centre embedding [Elman, 1991a]. This growth is mainly due to the fact that the SRN represents the recursive behaviour as a finite state machine (FSM). To do so, it needs an exponentially growing amount of neurones with respect to the number of states in the system

Kohonen's Self-Organising Feature Maps

During all of the 1970's, Teuvo Kohonen continued working on his associative memories and his self-organising feature map. Unlike back-propagation which was a supervised learning algorithm, the Kohonen learning rule does not involve supervision. Thus it can discover clusters of dependencies in unstructured data sets whereas the data input for back-propagation algorithm needed to be structured.

The Kohonen network is known to implement a vector quantisation algorithm, well suited for clustering purposes. Originally, this model was an abstraction from the visual model presented by Von der Malsburg in 1973. Kohonen eliminated the interneuronal weight modifications during the training phase and simplified the adaptation of the weights. The feature map is an abstraction of the biological topology preserving maps found in the human visual system [Kohonen, 1982a-c, 1984, 1988, 1990a-b] [Malsburg, 1973].

Kohonen defines a one layer map, where all neurones are connected to the same set of input fibres. After determining the best match for an input vector, the weights of the input fibres (or synapses) of neurones within one region are changed according to the input excitation. By defining a horizontal inter-neuronal structure, lateral inhibition (used to determine the best match) is implemented. In certain cases, the system self-organises, whereby regions on the map are formed, representing data falling within a certain statistical cluster, and thus automatically forming categories (dependent on the internal coding of the input patterns). By doing so, a highly efficient, powerful statistical classifier is obtained.

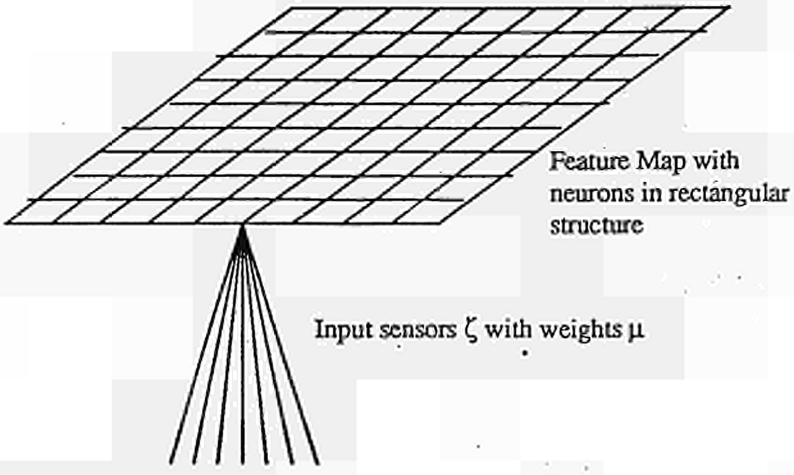


FIGURE 3.10: THE KOHONEN FEATURE MAP

Kohonen feature maps implement a number of important properties:

- First, the feature map conserves the topology (that is, if two object are close to each other in the probability space, they will also be close to each other after the mapping) of the original probability distribution function of the data set, even after a dimension reduction.

- Next, feature maps implement an automatic feature selection. It does not matter whether one adds multiple redundant input sensors, the data is only organised on the relevant features given the context in which they occur.

- Finally, the feature map organises on frequency as well as on context. That is, the frequency of a certain input pattern is just as important as the overlap between parts of these patterns. By doing so, feature maps implement a map of conditional probabilities.

The Kohonen formalism is a competitive learning algorithm. A two-dimensional map is constructed in a rectangular or hexagonal structure of individual neurones. Each neurone *i* has a number of input sensors with an input activation ξ_j and an input weight μ_{ij}. All neurones have the same number of input sensors and input weights. The activation of neurone *i* is calculated by:

$$y = \sum_{j=1}^{n} \mu_{ij} \cdot \xi_j \qquad \text{(EQ 20)}$$

Actually, this is not an activation function as we know it from the feed-forward networks. In this case the activation value is not used to feed-forward an activation value to the input of a following layer, but it is used to determine a measure of correlation between the input activation and the weight vector of a neurone. The neurone (or area) best representing the input activation can be determined by finding the neurone with the highest output activity. This might be considered as the neurone *r* with the best match for input vector *x* for all neurones, or the minimum Euclidean distance between the vectors *x* and *w* [3].

$$\|x - w_r\| = \min_i \|x - w_i\| \qquad \text{(EQ 21)}$$

The learning rule operates in the following way:

First, copy the activation values of an input vector *x* into all input activation sensors of all neurones. Next, determine the best match by finding the neurone with the minimum

[3] ξ and μ indicates *sensor values* of input values and weights. *x* and *w* indicate *vectors* of input values and weights.

mathematical distance between input and weight values (this neurone can be referred to as *Best Matching Unit* or *BMU*). Then, adapt the weights of the neurones within a certain region of this minimum, so they'll recognise the current input vector better in the future. A general learning function is [Kohonen, 1984]:

$$\frac{d}{dt}\mu_{ij} = \alpha(t)\left[\eta_i(t)\xi_j(t) - \gamma[\eta_i(t)]\mu_{ij}(t)\right] \qquad \text{(EQ 22)}$$

where α (t) implements a decreasing function in time in order to guarantee convergence, η (t) is a function of the distance of the neurone to the BMU, and g represents some non-linear scalar function of η (t). This rule adopts the weights of the neurones in the neighbourhood of the BMU.

By doing so, the BMU and the neighbouring neurones all represent the input vector better after the weight update and it will be recognised earlier in future situations. Because the area surrounding the BMU is also updated, very interesting (recurrent) neighbourhood effects occur during the training process, which enable the model to derive a topological map by self-organising means. Without the neighbourhood effects, the model would implement a much less sophisticated mapping.

If neurones are updated within a certain area of the BMU *c*, and one takes $\eta_{i}(t) = 1$ inside the area and $\eta_{i}(t) = 0$ outside the area, $\gamma(0) = 0$, and $\gamma(1) = 1$, then this equation can be rewritten as:

$$\frac{d}{dt}\mu_{ij} = \alpha(t)\left[\xi_j(t) - \mu_{ij}(t)\right] \qquad \text{(EQ 23)}$$

Research carried out by [Ritter et al., 1989a] showed that "bell-shaped" functions perform best for $\alpha(t)$.

Rewriting these equations in vector format with *w* indicating the weight vector and *x* the input vector results in the following formulation of the learning algorithm:

- Initialise all weight vectors *w* with random values between 0 and 1.
- Iteratively consider different inputs, and determine the neurone *s* for which the best match between the input values x_S and weight values w_S among all neurones *r* in the feature map:

$$\forall r(\|w_s(t) - x(t)\| \leq \|w_r(t) - x(t)\|) \qquad \text{(EQ 24)}$$

- Update all weights according to the Kohonen Learning rule:

$$w_r(t+1) = w_r(t) + \varepsilon(t) \bullet \Phi_{rs}(t) \bullet (x(t) - w_r(t)) \qquad \text{(EQ 25)}$$

where:

$$\Phi_{rs} = e^{-\frac{\|r-s\|}{(2\sigma(t))^2}} \qquad \text{(EQ 26)}$$

|| r - s || is the physical distance between neurone r and s on the map, and

$$\varepsilon(t) = \varepsilon_{\max} \bullet \left(\frac{\varepsilon_{\min}}{\varepsilon_{\max}}\right)^{\frac{t}{t_{\max}}} \qquad \text{(EQ 27)}$$

$$(t) = \sigma_{\max} \bullet \left(\frac{\sigma_{\min}}{\sigma_{\max}}\right)^{\frac{t}{t_{\max}}} \qquad \text{(EQ 28)}$$

$$\varepsilon_{\min} \in [0,1], \varepsilon_{\max} \in [0,1], \sigma_{\min} \in [0,1], \sigma_{\max} = \frac{\sqrt{N}}{2} \qquad \text{(EQ 29)}$$

where *N* is the total number of neurones in the feature map.

Equation 25 is derived from equation 23 by substituting two bell shaped functions Φ and ε(t), for α (t). After numerous cycles, a topological map will be formed, holding related elements in neighbouring regions.

A self-organising process in time is given on the next page. Here, a two-dimensional feature map with two-dimensional sensors (ζ_1,ζ_2) is used to map a homogeneous distributed set of input vectors from $\mathbf{R}^2$ to $\mathbf{R}^2$ (similar plots can be made for mappings with different dimensions).

The neurones can obtain all possible values in the domain [0,1]. The two dimensions in the training vectors can only obtain values from the set:

$$\{\frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, \frac{7}{7},\} \qquad \text{(EQ 30)}$$

The training set contains one element of all possible vectors in this domain. That is, the training set holds the following vectors:

$$\{\frac{1}{7}, \frac{1}{7}\}, \{\frac{1}{7}, \frac{2}{7}\}, \{\frac{1}{7}, \frac{3}{7}\}, \ldots, \{\frac{7}{7}, \frac{7}{7}\}$$

During the training session, vectors are selected randomly from this set. Every factor has an equal probability to be selected. Therefore, the probability distribution of the input space is considered homogeneous.

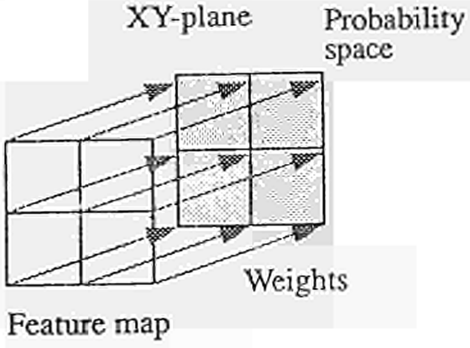


FIGURE 3.11: IN THE SELF-ORGANISING STATE, THE WEIGHT VECTORS OF THE FEATURE MAP REPRESENT SPECIFIC POINTS IN THE PROBABILITY SPACE.

By plotting the values of all weight vectors in a XY-plane, and by connecting each neurone with its direct neighbours, a perfect rectangular graph is obtained if and only if the feature map reaches the so-called self-organising state. In this state, the feature map is perfectly ordered, that is: all neurones represent data that have a minimal Euclidean distance to all its neighbours. In that case, only one solution is possible, the graph forms the same map as the original topology of the feature map, in this case a perfect rectangular shape.

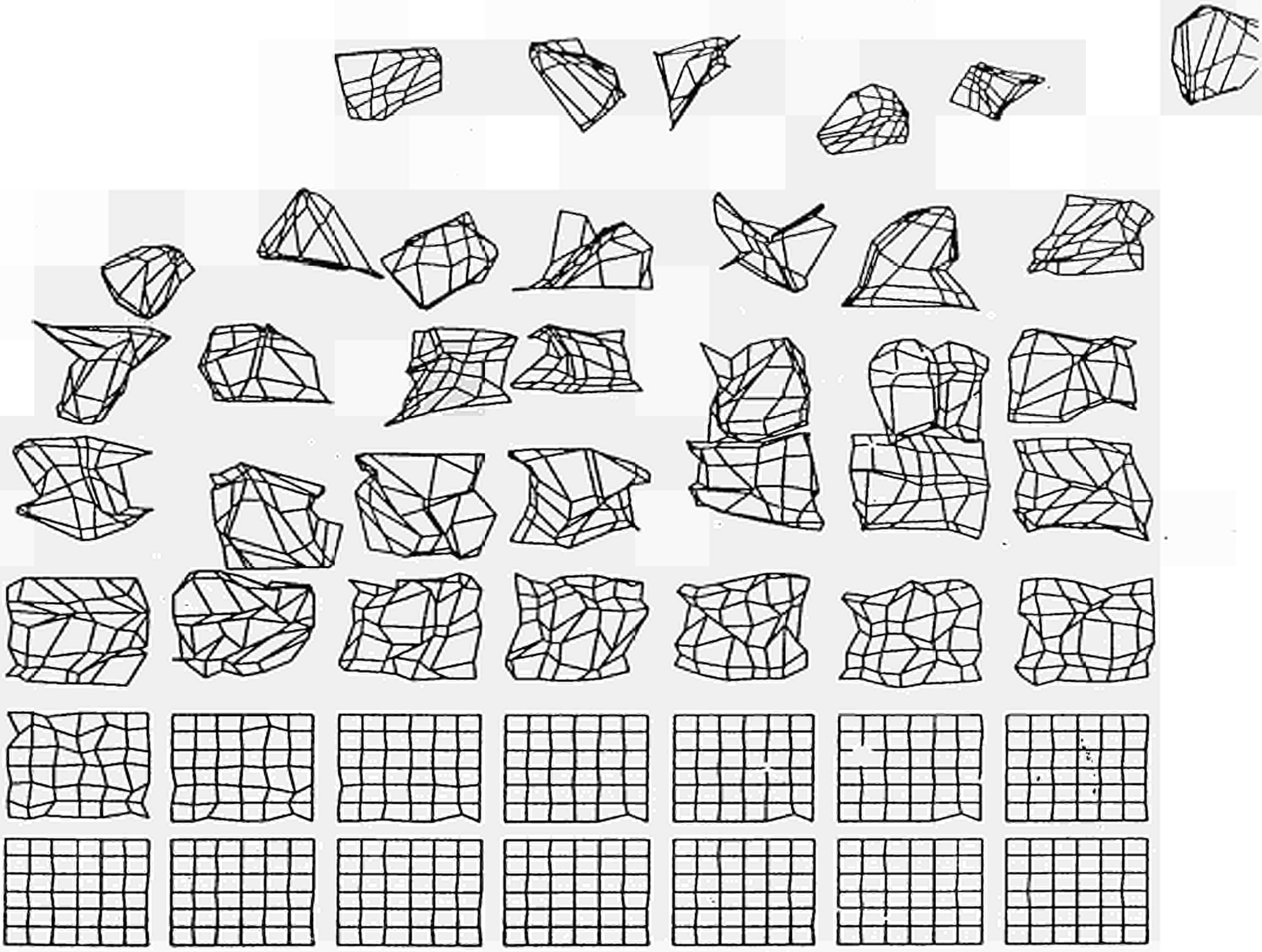


Figure 3.12: The Self-Organising Process. From random weights (upper-left corner) to the self-organising state (bottom-right corner).

The Hypermap

The Hypermap is a more context-sensitive variation of the feature map. It was introduced by [Kohonen, 1991]. The normal feature map is not particularly context sensitive, making it hard to use in some cases. In the hypermap training algorithm, the data is presented within its (natural) context as a concatenated vector of the object data x_O and the context data x_C.

$$x'(t) = [x_o(t), x_c(t)] \qquad \text{(EQ 31)}$$

$$w'(t) = [w_o(t), w_c(t)] \qquad \text{(EQ 32)}$$

The context can be a shifting window as well as a more artificial notion of context (like a context category). First, the feature map is trained on this context only by using the regular training rule. During this first training phase, the object weight values stay fixed.

$$w_{cr}(t+1) = w_{cr}(t) + \varepsilon(t) \bullet \Phi_{rs}(t) \bullet (x_c(t) - w_{cr}(t)) \qquad \text{(EQ 33)}$$

After a certain saturation value is reached, the context weights are fixed and the map is trained on the object sensor values.

$$w_{or}(t+1) = w_{or}(t) + \varepsilon(t) \bullet \Phi_{rs}(t) \bullet (x_o(t) - w_{or}(t)) \qquad \text{(EQ 34)}$$

In doing so, the global properties of the data distribution are captured first, while the details are determined later.

It appeared that these maps showed a 10-20% performance increase in speech recognition applications. Moreover, this adapted training algorithm performs much better in large feature maps (over 1000 neurones) because they get enfolded or tangled less easily.

The Semantotopic Map

In [Ritter et al, 1989b] and [Ritter et al., 1990] sentences are presented to the system as a vector concatenation of words (X_S) with their corresponding contextual structure (X_C). Representing single words without context has no meaning in the Kohonen model. The assigned codes are arbitrary, and therefore of crucial influence on the derivation of organisation in the map: by assigning another code to an object it is placed on a different position of the map. However, if a semantic map can be derived by showing the words in their proper context, then the individual encodings become irrelevant.

Although the Kohonen model offers interesting results in language acquisition, it does not provide a complete model for language acquisition because of the inability of the model to derive and to process language structure.

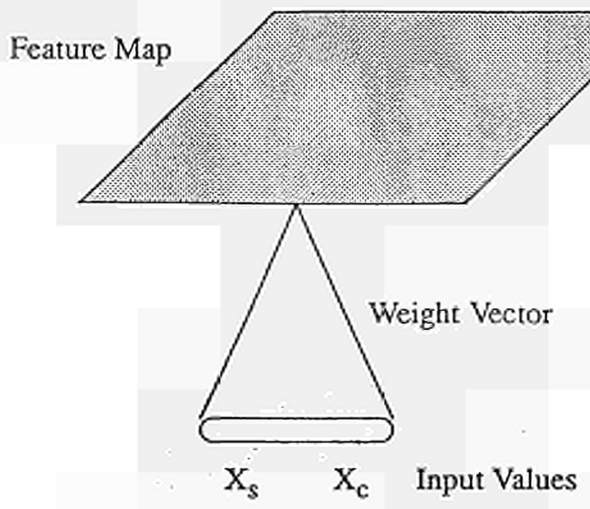


FIGURE 3.13: SEMANTOTOPIC FEATURE MAP.

[Jagota, 1990] shows another associative memory application of neural networks: a lexicon is implemented in a Hopfield Network. The known advantages of neural networks, such as incomplete retrieval, error correction and generalisation were all observed.

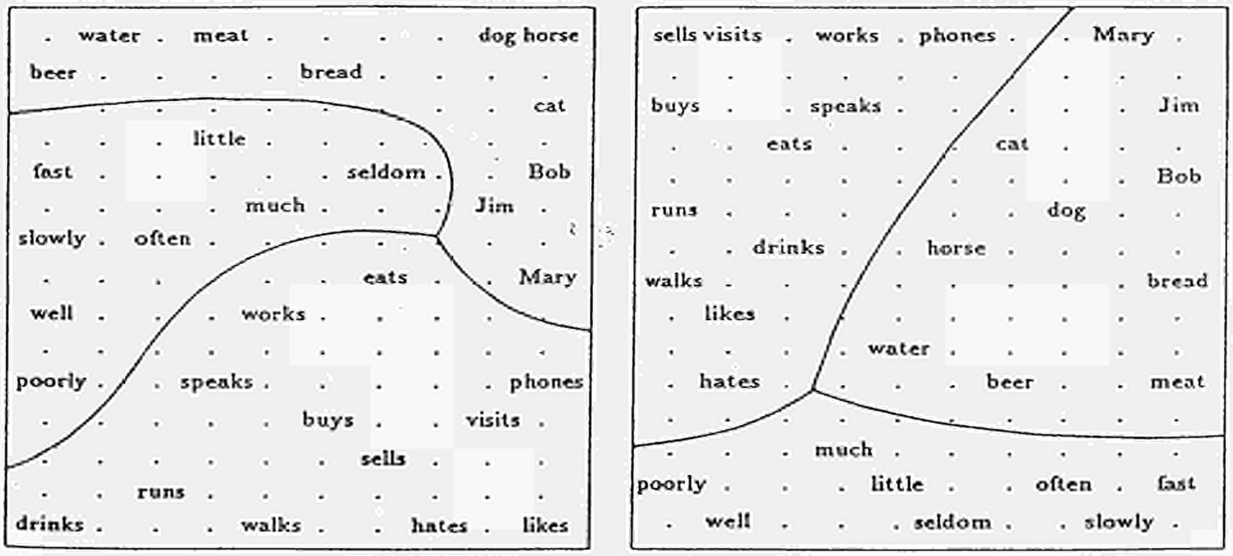


FIGURE 3.14: THE SEMANTOTOPIC MAP FOR INPUT MANUALLY TAGGED WITH CONTEXTUAL INFORMATION (LEFT) AND THE MAP FOR A RESTRICTED CONTEXT OF THE IMMEDIATE PREDECESSOR ONLY (RIGHT) (REPRINTED FROM [RITTER ET AL., 1989B]).

Related to the recurrent neural networks described above, is the problem of language generation. [Kamimura, 1990a-b] shows that a recurrent neural network can generate arbitrarily long sentences, where [Williams et al., 1989a-b] has shown instabilities. A variable learning rate is added to the back-propagation algorithm, and the Minkowski-r power metrics are used instead of an ordinary error function. The sentences remain however simple (finite state grammars). [Kukich, 1988] shows the relevance of connectionism in language generation. Input is fed into a time-delay network (also called running text in an input window). By back-propagating input/output pairs linguistic transformations, sememe-to-morpheme and morpheme-to-phrase transformations are learned correctly by the network.

Adaptive Resonance Theory

Stability and plasticity.

If we do not know beforehand what the correct categorisation for a pattern is, or if we want to use the structure that is implicit in the input space itself, we can let the data decide. The mapping will adapt itself to the peculiarities of the domain. In neural network terms this is called self-organising adaptation. This learning paradigm for neural networks usually takes the form of so called competitive learning. The units in a layer of neurones compete for the best match to the input patterns, i.e. the highest activation. This is determined by their current weights. A unit that wins the competition may then adjust its weights so that it will match the same pattern even better in the future. Clusters in the input space will activate the same units, so that one can speak of category formation.

This scenario works just fine if the input space is in some way fixed, or if the number of nodes available for categorisation is unbounded. Limits on resources or adaptation to a highly dynamic environment however, introduce new problems one has to face when designing neural information processing systems.

These problems go under the name of the stability- plasticity dilemma. In the journal *Biological Cybernetics* Stephen Grossberg [Grossberg, 1976a,b] first addressed these problems in a series of two related articles. In the first of these two he demonstrated the problem by a simple hypothetical sequence of pattern presentations. When the input environment changes, one would like to code the new patterns (plasticity), while at the same time retaining the old codes (stability). Because classical competitive learning is in no way buffered from change, except by shutting the learning process down, it fails on this point. The only solution seemed to have a omniscient external process which could monitor the progress of learning and reorganise the categories off line.

In the second article he offered a solution for this problem: Adaptive Resonance Theory (ART). This is a theory about how a network architecture can self-organise stable recognition codes in real-time in a spatially and temporally dynamic environment.

The theory has implications on two different sides. The neurobiological foundations, which are beyond the scope of this text, and the application of ART architectures as an extension of neural technology. For a long time, the work of Grossberg and his colleagues has not found widespread recognition. This is partly caused by the biological orientation of their papers, and partly by their terminology which has developed since the late 1960's, but has not adapted to the terminology of the neural and connectionist revival of the 1980's.

Nonetheless, more recently ART has seen several variants of implementation (ART1, ART2, ART2A, ART3, ARTMAP, Fuzzy ARTMAP) which are being applied to various tasks. We will return to the discussion of their workings, their differences, their applicability and the related pros and cons below. First the ideas behind ART in general will be presented.

The ideas behind ART

An ART module consists of two parts: an attentional subsystem and an orientational subsystem. These collaborate in the categorisation process. The attentional subsystem has two layers of units, connected by weights, which do the actual categorisation and learning of patterns. The orienting subsystem directs the dynamics of the network to ensure stability and plasticity. The key concept here is that of vigilance. According to a vigilance parameter of the orienting subsystem, the network decides whether a pattern is new and should receive a new category code, or familiar enough, and should be represented by an existing category.

The input patterns are presented to the network at the first layer (traditionally called F1). Through the bottom-up weights the pattern causes activation of units in the second (categorisation) layer, F2. The code for a category can be distributed across several F2 units, but usually just one unit is selected (winner-takes-all). The code from F2 sends signals through the top-down weights towards F1. These represent the prototype of the selected category. Here the orienting subsystem comes into play. There are now two options. If the top-down prototype template matches the input pattern well enough according to the vigilance parameter the two layers can resonate. During this resonance learning occurs, the weights are adjusted between co-active units. If on the other hand the match is too poor, the orienting subsystem causes a reset of the chosen category. It will inhibit the responsible F2 units for the duration of this particular pattern. Bottom-up activation can then select another unit as the winner, continuing until resonance occurs. If none of the present codes suffice and there are still resources available, a new code (unit) will be allocated to this pattern. This is illustrated below by an example from ART1.

Implementation

Adaptive resonance modules were originally specified as a complex system of dynamic network equations [Grossberg, 1976b]. When applying ART as a realistic architecture for a categorisation task, a number of implementation choices have to be made. An example: the winner-takes-all choice in the categorisation layer is neurally feasible by convergence of a mechanism of lateral inhibition. In a computer simulation it is however much easier to just pick the highest activation from the F2 layer right away. Other choices are the implementation of the matching process, and of the category reset by the orienting subsystem. Simulating the system network equations would be computationally too expensive, but by making certain simplifying assumptions about the nature of the inputs (e.g. binary vs. analogue) or the nature of the desired category codes (e.g. local vs. distributed) good results are obtained.

The main subdivisions between ART networks will be discussed now. Most of the relevant original articles have been reprinted in [Carpenter et al., 1991c]

ART1

This architecture was introduced in [Carpenter et al., 1987a]. It is designed for use with binary input patterns. The resulting categories are of the winner take all type. The architecture is illustrated in the figure below: ART1 module.

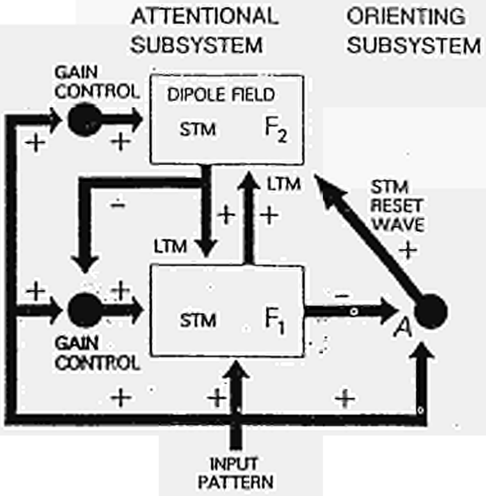


FIGURE 3.15: ART1 MODULE (REPRINTED FROM: [CARPENTER ET AL., 1991C]).

Because of the binary input patterns ART1 can make use of the so called 2/3 rule to match bottom up data with top-down learned prototypes. The F1 layer receives not only external inputs and top-down signals but also a non-specific signal from a gain control node in the orienting subsystem. Two out of the three of these inputs have to match in order to achieve resonance. The 2/3 rule allows for the priming of F1 by top-down expectations in the absence of external inputs.

In absence of instant resonance the network has to find a new code. The search for a correct F2 unit is illustrated in the following figure.

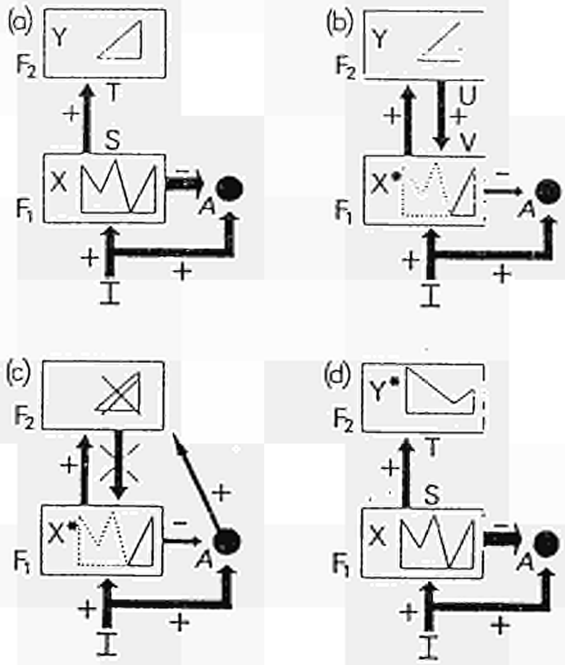


FIGURE 3.16: SEARCH FOR A CORRECT F2 UNIT IN ART1 (REPRINTED FROM: [CARPENTER ET AL., 1991C]).

The reset wave upon mismatch is mediated by an arousal node A. It becomes active when the 2/3 rule indicates a measure of similarity that is lower than a pre-set vigilance parameter (usually denoted by rho). If this parameter has a high value a new category is chosen faster. In this way one can control the granularity of the categories.

In the actual implementation the reset categorisation nodes are simply excluded from the rest of the competition. The resulting categories are represented by just one node in F2 (local representations) and the categories are non-overlapping. One can choose between slow and fast learning. In the first case the weights are adjusted gradually. In the fast case a pattern is learned in “one shot”.

A problem with the ART1 network was that it could not distinguish patterns which are embedded within other patterns; subset and superset patterns. To solve this problem a Weber rule and active decay for the weights was introduced. The Weber rule forces the weights that code a "smaller" pattern to get a higher value. The principle of active decay is that weights from F1 to the winning node in F2 that are not used will progressively decay.

A number of formal proofs in [Carpenter et al., 1987a] establish the following properties of ART1.

- The 2/3 rule is sufficient for the stable learning of arbitrary binary input sequences.

- The search mediated by the reset wave proceeds in the optimal order.

- Once learned, a category can be accessed directly without search.

ART2

This variation was introduced in [Carpenter et al., 1987b]. It is designed for use with analogue input patterns. Just as in ART1 the resulting categories are of the winner take all type. The architecture is illustrated in the following figure.

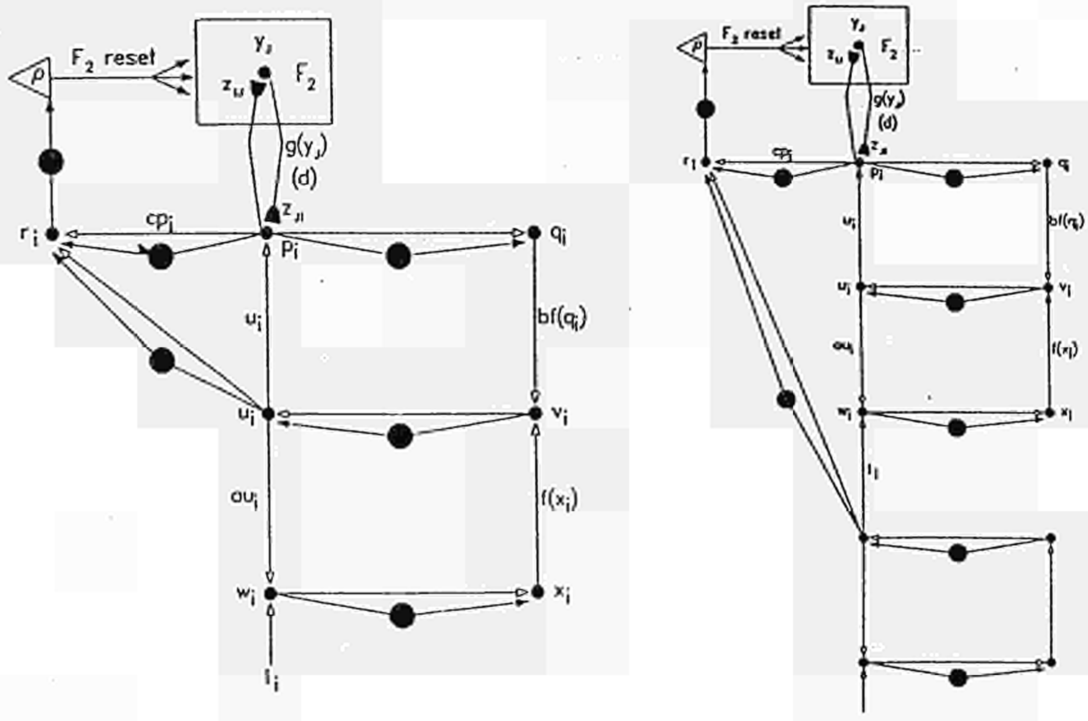


FIGURE 3.17: TWO ALTERNATIVE ART2 ARCHITECTURES (REPRINTED FROM: [CARPENTER ET AL., 1991C]).

Binary input patterns always differ by at least 1 full bit. A 2/3 rule suffices. Analogue patterns however pose some new problems. They may differ by arbitrarily small numbers, have various baseline intensities, and contain distortions by noise. The architecture of ART2 and its variants is largely similar to the ART1 design. The main difference is the structure of the F1 layer. In ART2 this layer has to do the pre-processing and normalisation of the external input, i.e. remove noise and bring down to one scale, and also the matching of the prototypes with this pre-processed input. Of course the input itself must be buffered in some way as to prevent overriding by the top-down expectations. In order to fulfil all these tasks the F1 layer of an ART2 network usually consists of three slabs, each with two subdivisions. In the first slab the inputs are buffered, in the third slab the top-down weights are buffered, in the middle slab they are both normalised and matched. The principle of operation is the same as for ART1, but the actual mechanisms used are far more complex.

[Carpenter et al., 1991a] defines an purely algorithmic version of ART2, called ART2-A, which runs up to three times faster as the original version.

ART3

Already in ART2 there were a lot of parameters and corresponding degrees of freedom, more on which below. The ART3 architecture [Carpenter et al., 1990] is even more complex. It uses multiple mechanisms from the F1 layer in ART2 throughout the whole network. It is designed for use with analogue input patterns and allows the formation of distributed, partially overlapping category codes. The principles of function are again the familiar ones of ART in general. Besides that, ART3 introduces a new level of memory, that of habituating transmitter channels to implement the search for distributed code. To our knowledge it is not very widely used for real-world applications as it is computationally quite heavy. The distributed category codes might however be needed in multi-module hierarchies.

ARTMAP

Unlike the previous ART networks, ARTMAP [Carpenter et al., 1991b] is an architecture for supervised learning. Unlike the backpropagation algorithm ARTMAP learns by matching input-output pairs, not by error correction. The basic idea is quite simple. See the figure below (Block diagram of an ARTMAP system). Two coupled unsupervised ART modules (ART(a) and ART(b)) form internal representations of respectively input and output. An extra module (map-field) associates the two codes, enabling an arbitrary mapping between two vector spaces of patterns. The learning proceeds as follows. Patterns are presented to both

ART(a) and ART(b) input layers. If a mismatch occurs in ART (b) between the prototype output of the map-field and the target pattern a process called "match tracking" increases the vigilance parameter of the input module, ART(a), by the minimal amount needed to search for a new recognition category. In this way ARTMAP is capable of learning to distinguish rare but important events from frequent events that are perceptually similar but that predict different consequences.

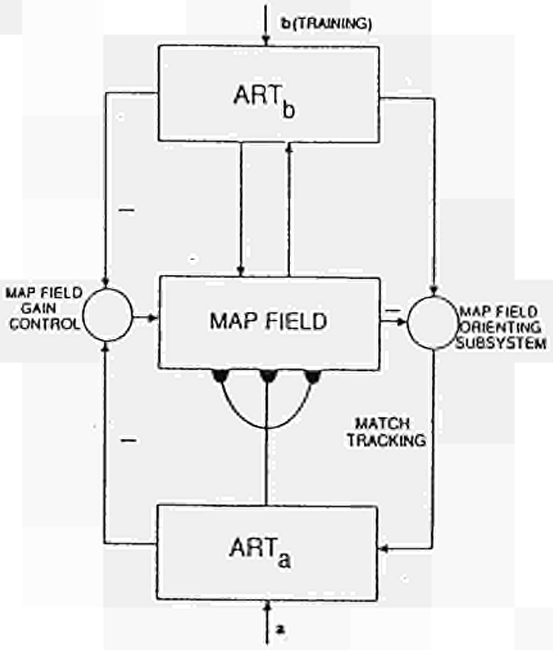


FIGURE 3.18: BLOCK DIAGRAM OF AN ARTMAP SYSTEM (REPRINTED FROM: [CARPENTER ET AL., 1991C]).

[Carpenter et al., 1991b] claim the following results:

- Stable learning of one ore more non-stationary patterns sets.
- Learning orders of magnitude more quickly, efficiently, and accurately than alternative algorithms (among which backpropagation).
- Tested on a machine learning benchmark ARTMAP achieves 100% accuracy after training on less than half the input patterns in the database.

How useful is ART?

The family of ART networks seems to be an interesting solution to many problems of other neural network approaches, most notably stability and plasticity of both competitive learning and backpropagation. Because the learning process is self-controlled, far fewer patterns and iterations over patterns are needed in order to get satisfying results. So theoretically speaking ART networks are faster. In their purest form however, they are specified as complex systems of network equations, i.e. not in an algorithmic fashion. The system with its many parameters must be simulated on a serial machine. Although the required number of pattern presentations may be much lower, it is uncertain how the actual total training time relates to that of other neural networks. This is of course very dependent on the implementation details.

ART networks have, because of their neuro-biological foundations, the potential for efficient parallel hardware implementation. This is due to the fact that all processing in the network only depends on information that is locally available to each unit.

The applications of this architecture to technological problems, are not yet widespread. This is partly a cultural phenomenon, but also the result of some of the weak sides of ART. The more complete versions of ART, like ART2 and 3, are quite complex, with many parameters to tune. Besides this, the simulation algorithms are computationally expensive. More research ought to be done on issues like scalability, parameter robustness, and efficient implementation. Still, Adaptive Resonance Theory seems quite promising.

Neuronal Group Selection and Genetic Algorithms

Even more biologically inspired than Linsker, Grossberg and Von der Malsburg, is the work of [Edelman, 1987] and [Reeke et al., 1988]. Their Neuronal Group Selection (NGS) theory applies a genetic approach to the formation of maps on the cortex and the growth or change of neural connections, which constitute the learning processes of human beings. Their models have learning rules that model populations of organisms as populations of distributed patterns within neuronal groups. Each neuronal group represents a solution to the problem. Better solutions (e.g. groups) multiply (or grow) faster than worse solutions, resulting in a good (or optimal) solution after a few generations.

This biologically and evolutionary inspired theory states that human genes do not have enough memory capacity to encode the structure of the human nervous system before it actually develops. This, together with the fact that even twins have completely different nervous systems, implies that there has to be some way of "competitive growing" or, in other words, some kind of Darwinist process that eliminates structures that are irrelevant or unsuccessful (according to some fitness function).

Edelman and Reeke criticise connectionism for avoiding the selectionist aspect in its current models. Their strong point is the clear absence of identical neuronal structures in nature: even twins exhibit different neuronal structures. One can compare this with the work done by

[Goldberg, 1989] on genetic algorithms, that are much used for the optimisation of back-propagating neural networks.

Hybrid Models

Another variation seen in different research work is the level of connectionism. Connectionist solutions may not always be best as stated earlier. One can decide to use symbolic methods to solve sub-problems for various reasons, like complexity and performance. Accordingly, literature provides us with a number of hybrid solutions in neurolinguistics.

Modules may be replaced by symbolic methods, e.g.: a lexicon for automatic feature detection and concept representation in [Miikkulainen et al., 1988a-b]. Another solution to avoid the pre-wiring and pre-categorising disadvantage of back-propagation can be found in [Hendler, 1989]. The addition of a conventional parser as pre-processor is used in [Waltz et al. 1984, 1985]. Even more obvious are the implementations of symbolic methods in connectionist systems. [Touretzky, 1987] tackles the slot filler problem by using a connectionist knowledge representation scheme. [Dolan et al., 1987] proposes a scheme for implementing schemata in a connectionist network. [Touretzky, 1986] implements BoltzCONS (a combination of a Boltzmann Machine and the basic list construction operation CONS in LISP), a recursive mechanism resulting in a connectionist production system [Touretzky et al., 1988]. Even Connectionist expert systems can be found [Saito et al., 1988], [Gallant, 1988], [Bounds, 1989], [Bradshaw et al., 1989], and [Gutknecht et al., 1990].

This list becomes longer every day, mainly due to the infinitely possible combinations of many conventional symbolic AI and new connectionist techniques.

The State of the Art

As of the summer of 1992, there is a split within the neural network research community. On the one hand, the more biologically inspired research group started organising its own conferences in computational neuroscience and behavioural sciences. On the other hand, there is a growing group of people who are trying to relate traditional statistical pattern recognition and neural networks. This second group is more interested in abstract models such as the back-propagation model and Kohonen feature maps.

At this very moment neural network research is still flourishing, although it has lost some of its magic over the years as the working of the algorithms became more clear and less exciting. Currently there are two International Joint Conferences on Neural Networks (IJCNN) every year (one in the US and one in the Far-East). Each of them still attracts more than 2,000 participants. Europe has its own large conference in the form of the International Conference on Artificial Neural Networks (ICANN) which was about the same size as each of the IJCNN's. In addition, large international Artificial Intelligence conferences such as the IJCAI, ECAI, SPIE, COGSCI, ICML, and AAAI have important neural network sessions. It seems that neural networks have established a much stronger position in the research community than they had in the 60's. This time they will probably remain, at least for their unrivalled power in applications such as hand-written Optical Character Recognition and the prediction of non-linear time series such as sun spots and stock market prices.

Commercial applications of neural networks can be found everywhere. In some cases (such as OCR and hand-written recognition) they have even beaten all other traditional techniques.

3.4 Radical Connectionism

Artificial neural networks are much less sophisticated than the human nervous system. It is possible to derive a number of basic features that provide the specific properties of neural networks as they are referred to by many researchers these days.

If a neural network has the following properties:

- Massive parallelism,
- Natural data input,
- Distributed data representation, and
- Self-organisation

> *only then* can one refer to the advantages of biological neural networks such as adaptive behaviour, implicit generalisation, error-correcting capabilities, and fault-tolerant processing.
>
> *If* on the other hand, one of these four features is missing, one should be extremely careful referring to the typical characteristics of biological neural networks.

Mainly responsible for the generalising capabilities is the distributed data representation. However, if the representation does not use a massively parallel network, it will not be good enough to provide generalisation in all cases. If a neural network does not implement any form of generalisation it will be equal to a lookup table and therefore not worth the classification neural network. The same holds for the fault-tolerant processing and error-correcting capabilities of neural networks, which are in a way nothing else than generalisations into the correct data sets.

The adaptive behaviour of a neural network is mainly due to a (self-organising) learning rule that generalises over known cases and that classifies new cases in known and new categories. If one does not use natural data input, a (manual) preclassification is carried out in which important information gets lost. One of the main properties of self-organisation is that it performs automatic feature selection, given the context in which an object occurs. By artificially labelling an object, it can only be classified or organised on the basis of this artificial labelling. Therefore natural data input and self-organisation are essential properties of artificial neural networks. In [Dorffner, 1991] the author mentions the four points above as properties of "radical connectionism", as opposed to the more moderated forms of connectionism. Here it is rather seen as "rather "essential" in every connectionist modelling project that implies typical neural behaviour or pretends to be psychologically plausible.

3.5 Expected Problems

On the one hand, neural networks seem to be able to outperform symbolic methods in various ways. On the other hand, many problems in practical neural network use remain unsolved. This section will discuss some disadvantages of neural networks.

Precise Computations, Dynamic Binding & Hierarchical Structures

First, neural networks are poor at precise computations. Sequential computers will outperform neural networks in these kind of computations. This shortcoming of neural networks might not be a problem in natural-language processing, because precise computations may not be needed there.

Second, symbolic reference, dynamic binding and hierarchical structures are essential elements of all cognitive theories. There should at least be a functional equivalent for these mechanisms in neural networks. However, the most significant problems for neural research to solve are the dynamic binding problem and the mapping of hierarchical structures to vectors.

The first problem directly results from the fact that distributed neural networks are relatively unstable memory elements for exact data. One cannot store a variable in a particular place for an unlimited time. The second problem is a special case of the first problem. Due to the instable memory elements, recursive and hierarchical structures cannot be stored for an unlimited time.

Natural language is one of the problems in which structure is explicitly present by means of syntactical structures. However, neural networks are only capable of processing vector elements. So, how does one map these hierarchical structures to vectors without loss of information? Even more important, how does one map the vectors back to a hierarchical structure?

Both questions are not solved yet. Once they are, the sky will be the limit for the application of neural networks in natural language processing as well as other problems involving some form of learning and processing of hierarchical structures.

Temporal Sequences

In a sequential computing paradigm, sequences are implemented implicitly by the data flow. In parallel systems however, one has to implement a special mechanism to keep track of sequential dependencies. Neural networks do not appear to be good in representing changes in time. It is quite easy to develop a neural network that realises some sort of classification function, but the implementation of time dependency is another story. There are a number of time influences in decision making: e.g. time- varying responses (long term by changing the weights of the interconnections; short term to represent time in a network) and sequence (analysing inherently sequential input). Overall, three solutions can be given to the short-term timing problems and the sequence handling. Most simple is the addition of extra input bits, which represent the timing information explicitly. This solution does not have any biological plausibility and it results in network patterns that are too complex.

A second method is the construction of extra layers in the network with memory functions: so called feedback (or recurrent) loops, in particular popular in back-propagation [Jordan, 1986a-b] [Elman, 1988]. In principle, all self-organising models implement a form of recurrence by adopting their connection weights through local interaction with their neighbours only. Addition of other forms of recurrence quickly increases the complexity and may result in unstable models.

Another solution is a shifting window structure. Instead of using feedback connections to implement sequences, the data is presented in parallel to the neural network. By presenting the data within its natural context, the system becomes aware of sequences. The shifting window prevents the model from only processing fixed (or maximum) length data input. The model will never be able to develop an internal memory structure that represents data dependencies longer than the window size. Moreover, the model also implements a finite state machine [Ritter et al., 1990], [Sejnowski et al., 1986].

Already in the late 50s, a famous paper on the psychological plausibility of shifting windows was written by [Miller, 1956]. One of the main problems in interpreting this work is the question of what is shifted over: characters, words, sentences, or mental concepts?

Scalability

Artificial Neural Networks such as Kohonen Feature Maps and Back-Propagation are *not* scaleable. That is, if a certain problem can be solved for a small set of test data, this does not guarantee that the same problem can be solved for a larger data set.

There are a number reasons why this is the case:

- Larger data sets require more convergence time. Often, the time required to train a n times larger data set is of the order x^n.

- Larger data sets require larger neural networks. In the convergence process, the boundaries of the network are very important for stability (experiments with neural networks without boundaries (torso's) have proven to converge much harder). Larger neural networks have relatively less boundary neurones with respect to smaller ones (in a Kohonen feature map, the number of border neurones decreases with the order of the root of the total number of neurones). As a result, larger neural nets converge slower, or not al all...

For this reason one should be very careful with the interpretation of prototypes. A small prototype does not guarantee success with a large one. Even better, if large amounts of data are stored in the neural net, it can be guaranteed that the model will not work...

The Hypermap algorithm [Kohonen, 1991] and training algorithms such as incremental learning [Elman, 1991b] only extend the problem, they do not solve it.

General Criticism on Connectionist Models by Neuropsychologists

Various researchers in traditional artificial intelligence have plenty of criticism on the connectionist language processing paradigm, but then, it is exactly they who are attacked by the PDP group. In return, biologically oriented neurologists heavily criticise the neuropsychological plausibility of some aspects of the PDP models [Gigley, 1983, 1985].

First, PDP models include binary feature detectors and a mutual inhibition scheme to recognise input. From a neuropsychological viewpoint, this assumption is wrong. Only the recognised input is active, not the opposites or synonyms. According to Gigley, there is no inhibition of any information at the perceptual level. Therefore explicitly defining mutual inhibition is wrong.

Besides the psychological problems of lateral inhibition, there is also a practical one; all the inhibitory interconnections must be prewired by hand, an enormous amount of work.

This non-adaptive character of the back-propagating algorithm and the need to define all the micro features by hand, might convince the reader that self-organising techniques are better in natural language processing. Critique from Edelman and Reeke also points in that direction [Reeke et al., 1988].

Although this critique definitely applies to all artificial neural networks, one should be aware of the fact that NLP problems are complex by their very nature. Therefore, one can defend the position that one must be extremely careful in implementing the already complex NLP models in even more complex neural models.

General Criticism on Connectionist Models by Cognitive Scientists

[Fodor et al., 1988] advocate the position that neural networks are never stable enough to implement a dynamic binding system between variables (or symbols) and values. Without such a system, information on certain topics should be present in exactly the same form at different positions in the brain at exactly the same time. As this can never be the case, some form of symbolic reference process must be present in the human brain. As long as current artificial neural networks cannot model such a process, they do not model the human brain properly and should therefore not claim to be more psychologically plausible than current symbolic Artificial Intelligence techniques.

In other words, one needs to implement:

- Compositionality; the recursive combination of symbol structures in larger structures, and
- Distal access; a mechanism to refer a remote structure through some pointing device (needed by the compositionality).

Current neural networks are not able to implement (recursive) hierarchical structure without the loss of typical neural properties such as distributed data representation and automatic learning.

Part 2

The State-of-the-Art Report

4 Introduction to the State-of-the-Art

"Every time I fire a linguist my performance goes up"

-- Frederik Jelinek

Many different research projects will be reviewed in this part of the report. In this chapter three dimensions by which much of the current research can be categorised are presented. An intuitive indication of the expected degree of success of various approaches is given.

The connectionist approach offers a massively parallel, highly distributed and highly interconnected solution for the integration of various kinds of knowledge, with preservation of generality. It might be that connectionism or neural networks (despite all currently unsolved questions concerning learning, stability, recursion, firing rules, network architecture, etc.), will contribute to the research in information retrieval. Distributed data representation may solve many of the unsolved problems in IR by introducing a powerful and efficient knowledge integration and generalisation tool. However, distributed data representation and self-organisation trigger new problems that should be solved in an elegant manner.

Considering the general properties of neural networks, one can (intuitively) understand the common ground both disciplines share. Neural networks exhibit robust, adaptive, generalising and context sensitive behaviour in pattern recognition tasks comparable to IR. However, the application of neural networks in information retrieval will probably not be so easy as it seems at first sight. In the following chapter we will review examples of the application of neural networks in IR as found in the recent literature.

To provide some unifying themes to these, often very diverse, approaches we shall use three categorisation schemes for the discussion of the models in this chapter:

- Categorisation by Application
- Categorisation by Information Representation
- Categorisation by Retrieval Type

In the first scheme, models are categorised by their application. The following main groups have been determined: library management, clustering, interface design, user modelling, retrieving incomplete data, retrieving multi-media data, and information fusion. Within these groups there are a number of sub-groups that are expanded in the next paragraph.

Categorisation by Information Representation. In general, three types of data representation can be found in the neural models: locally distributed, sub-symbolic and fully distributed. The type of data-representation is very important for the capabilities of a model. Therefore, within each application category, the models will be ordered with respect to their internal representation scheme.

Applications can also be categorised according to the flavour of IR that they implement. On the one hand there is the situation where we have a large relatively static database and a very dynamic query. This is the situation most typical of normal IR. On the other hand we can have a dynamic information stream which must be matched against a more or less static query, such as a user profile. This is typical of information filtering or routing.

Other Reviews

More background information on the application of neural networks in information retrieval can be found in a number of review articles:

- [Doszkocs, 1991, 1992] and [Quast, 1992] describe the potential of artificial neural networks for self-organising and adaptive information representation and retrieval, offering new and complementary capabilities for dynamic information categorisation, generalisation, classification, feature extraction and learning. Potential applications in cataloguing, indexing and on-line searching in libraries are discussed.

- [Perez, 1991] and [Starks et al., 1991] provide the reader with a very short taste of the potential of neural nets for library management and analysis.

- [Rasmussen, 1991, 1992] discusses the application of parallel processing in information retrieval.

- [Widrow et al., 1994] gives an overview of commercial successful applications of neural network technology. One of the applications discussed is information management and optical character recognition.

Most interesting was the ELVIRA conference on advanced information retrieval techniques that was organised in the United Kingdom from May 3-5 1994. The program covered neural networks and fuzzy systems in information retrieval, retrieval using parallel computers, and the indexing and retrieval of multimedia objects. Because the proceedings could not be obtained, the exact contents of this conference have not been included in this report.

4.1 Three Dimensions of Categorisation

In this section we shall discuss the three categorisation schemes that we have chosen to distinguish the applications found in the literature in some more detail

Categorisation by Application-Type

Separated are the following application-types:

Library management:

- Serials management; the management of magazines and periodicals.
- Loan management; the management of loans to individuals and other institution. This also includes the prediction of demand and required title acquisitions.

Information clustering:

- Clustering of documents with respect to attached keywords, titles, or abstracts of full-text. As a result of the clustering, documents can be organised in related bibliographic categories.
- Clustering of concepts from large amounts of free-text; these concepts can be used for the derivation of bibliographic groups, thesauri, associated word lists, and interface design applications such as relevance feedback and user guidance.
- Associative memories & semantic networks; the (semi)-automatic linking of information by usage of hyperlinks. This linking on lexical, syntactical and semantical levels can be done by means of a clustering algorithm.

Interface Design:

- Hypertext; using a neural network as a spreading activation model to incorporate multiple sources or multiple types of information.
- Adaptive searches (query-answer pairs) on structured and free-text data bases; the generalisation and association of (previous) queries-answer sets. These techniques

can be used to generate more advanced queries from simpler ones, associate related queries, and complete incomplete queries automatically.

- Relevance feedback; the automatic incorporation of user feedback on the query results.

User Modelling:

- User modelling for current awareness and SDI; the derivation, maintenance, adaptation and retrieval of user interest models in time. In particular for filtering information out of large streams to implement SDI and current awareness applications.

Incomplete Searching:

- Fuzzy searching; matching incomplete data and incomplete queries. This is particularly interesting to cover up OCR errors, miswriting, foreign names and other error-sensitive data.

Searching in Multi-Media:

- Searching for multi-media data such as sound, pictures, and video by using multi-media queries.

Information Fusion:

- Case-based reasoning; combining different sources of information for intelligent systems, i.e. help desks and process control.

- Data base mining (information syntheses); deriving relations between data sets that are not clear at first side. Usage can be found in fraud detection, credit applications, and business intelligence.

Other Applications:

- Juke-box staging; the loading of the proper optical disk in a Document Information System (DIS) as a function of the user behaviour. This is a very complex dynamical system, that can be well predicted from automatically trained neural networks.

Categorisation by Information Representation Type

An important dimension of neural models is the structure of the data representation put on the input sensors. The first systems available tried many different types of neurones, training rules and network topologies, but they had one thing in common; they all used a local data representation. A second generation of connectionist IR systems could be characterised by a sub-symbolic data representation, whereas the most recent models use a fully distributed representation scheme. The chosen data-representation scheme is very important for the capabilities of the neural network.

In addition, a remark has to be made with respect to the internal encoding scheme of connectionist models in IR in general. Many neural net models are involved in natural sensor processing (for example frequencies in speech recognition, light intensities in vision, and movements in robotics). In connectionist IR, the sensor values are artificial, that is, they are assigned by some kind of lookup table: a symbolic object is translated into an artificial vector code.

Localist Information Retrieval Models

Connectionist systems with a local data representation are in fact nothing more than a complicated implementation of symbolic information processing paradigms. The local system behaves as a parallel lookup table and has nothing to do with what (biological) neural networks stand for.

The first connectionist IR models were based on local connectionist models. In general, most new neural models are developed in the context of connectionist Natural Language Processing systems. These models are often applied in IR by other researchers.

Localist systems are very limited in their abilities and were mainly evaluated for two effects that occurred: *spreading activation* and *lateral inhibition.*

In spreading activation, decisions are spread over time, so various knowledge sources can propose elements of interpretation. Two types can be distinguished: digital and analogue. Digital spreading activation can be found in [Charniak, 1983], where marker-passing algorithms run in parallel with the parsing and semantical processes, using a depth first search algorithm to select correct interpretations of objects. The analogue type involves a network of weighted associations where activation energy is spread over the network as a mathematical function of the strength of the interconnections [McClelland et al., 1981] [Rumelhart et al., 1982]. Both have an overkill effect. The digital form often results in too many search paths.

Analogue spreading of activations can result in activation of the entire network. Instead of using techniques like damping and decay (carefully chosen weights between units, so not the entire network is activated), the model uses lateral inhibition, as proposed in [Feldman, 1981] for modelling biological-vision systems.

Lateral inhibition prevents two opposing action systems to excite simultaneously. In parsing this means that nodes representing alternative analyses of the same input may not be activated at the same time. By connecting them with inhibitory connections, only one of them will survive when the network relaxes, so there will be no conflict. To be more explicit: the most likely one will survive. Nodes representing different lexical categories for the same word, nodes representing different senses of the same word, nodes representing conflicting case role assignments, and corresponding semantic or syntactic interpretations should be interconnected in a similar way.

Sub-Symbolic Information Retrieval Models

Already in the early days of connectionist modelling, one was aware of the limitations of local data representation. However, a training algorithm for multi-layer, fully distributed systems was not known yet. As the localist systems resembled much to symbolic data processing, the sub-symbolic models were somewhere in between of the symbolic models and the fully-distributed neural networks. The sub-symbolic model has an intermediate level of structure between the neural and symbolic levels. The sub-symbolic models all used hand-structured sets of (micro) features instead of flat (non-structured) vector input, which distinguished them from the fully-distributed models as discussed in one of the next sections.

The sub-symbolic paradigm was the next logical step from the study of symbolic systems to that of fully distributed processing. Much of the work done by the PDP group in the mid 80s could be defined as sub-symbolic information processing. It was the opinion of many that cognition is described at the sub-symbolic level, implemented by a connectionist model [Smolensky, 1987]. Their views were shared by other researchers like as Douglas Hofstadter.

In order to realise a coalition of nodes representing a consistent interpretation that will dominate after several interpretations, activation links are made between phrases and their constituents, words and different meanings, roles and fillers, and corresponding semantic and syntactic interpretations. Until the level of semantic interpretation, local representations are used.

By introducing micro-features of meaning [Hinton, 1981], distributed-knowledge representation schemes are added to the model. Micro-features have the powerful ability to define semantical concepts in terms of basic features that are shared with other concepts. So, meaning is represented by a pattern of micro-features, distributed over the network. Although the model does not perform learning and the model is not completely integrated, the use of lateral inhibition and micro-features provides a valuable addition to connectionist systems.

As mentioned in the introduction, information retrieval systems generally use no more than global surface properties. The need for a more content-sensitive methodology exists from the early beginning. A normal first step is the addition of semantic information to document collections. A next logical step is the combination of an inverted index with a semantical network, resulting in a hybrid model having the common integration problems known from symbolic AI. By designing an additional semantic network, the localist neural model implements some kind of hypertext system. This system can be seen as a sub-symbolic model.

Most of the systems that are investigated in the past use this representation scheme. However, non of them have been implemented commercially due to very limited scalability, trainability and processing power. Most sub-symbolic models are interesting from a psychological or philosophical point-of-view, definitely not from a real-world application point-of-view.

Fully Distributed Information Retrieval Models

The first fully distributed connectionist language-processing models could be observed at about the same time as the invention of back-propagation. In these models, the entire coding is distributed (*n* units, *m* concepts). The signals that are presented to the input of the model is no longer structured by hand as is the case in the sub-symbolic paradigm. By using such coding schemes, generalisation, association and robust behaviour become implicit features of the models.

It is claimed that these fully-distributed models are in more than one sense different from the symbolic as well as the statistical NLP models.

As local and sub-symbolic models are known for their implementation and maintenance problems, a shift towards the fully-distributed connectionist IR models can be seen in the early 1990s. IR is known to be a field of very large data bases. Therefore, the common connectionist ideas appeared to be difficult to implement as one can read in the following sections.

Categorisation by Different Approaches to Information Retrieval

Two main directions of neural network related research information retrieval can be observed.

- First, there are relatively static databases that are investigated with a dynamic query (free text search, also known as document retrieval systems).
- Next, there are more dynamic databases that need to be filtered with respect to a relatively static query (the filtering problem also known as current awareness systems and Selective Dissemination of Information, SDI).

In the first case the data can be pre-processed due to their static character. In the second case, the amounts of data are so large that there is no time whatsoever for a pre-processing phase. A direct context-sensitive hit-and-go must be made.

Early neural models adapt well to the paradigms currently used in information retrieval. Index terms can be replaced by processing units, hyperlinks by connections between units, and network training resembles the index normalisation process. The more recent supervised ANN models adapt less well to the notion of information retrieval. It is difficult to imagine what to teach a neural information retrieval system if it is used as a supervised training algorithm. The address space will almost always be too limited due to the large amounts of data to be processed. A combination of structured (query, retrieved document numbers) pairs does not seem plausible either, considering the restricted amount of memory of (current) neural network technology. Nevertheless, most of the neural IR models found in literature are based on these principles.

Also problematic are the so-called clustering networks. Due to the large amounts of data in free text databases, clustering is very expensive and is therefore considered irrelevant in constantly changing information retrieval environments.

More interesting is the unsupervised, associative memory type of models, that can be used to implement a specific pattern matching task. This type of neural networks can be particularly useful in a filtering application. Here, the memory demands of the neural network only need to fulfil the query (or interest) size, and not the size of the entire data base.

It is in this area where neural networks are expected to be most useful and relevant for information retrieval. Especially topics such as fuzzy retrieval, user-modelling in current awareness and SDI, concept formation and advanced interface design are within the scope of the project. Two categorisation schemes are used. First, the projects are ordered per application-type. The main categories used are serial- and loan management, clustering of bibliographic data and thesauri, interface design, filtering, retrieval of noisy data, retrieval of multi-media data, and data fusion. Next, per application type the models are organised chronologically and per data representation type, e.g. locally, sub-symbolic and distributed data representations.

It is obvious that in some cases these categorisation schemes cannot be followed strictly. In those cases, the best possible classes were chosen.

4.2 Expectations

What can be expected from the application of neural networks in information retrieval in a libraries context? Which applications do we expect to be successful and which not? Given the general properties of ANN's, without any in-depth research the following rough expectations can be stated on forehand with regard to the success of neural network applications:

Very Successful

- Retrieval of incomplete or noisy information
- Contents-based retrieval of multi-media information
- User modelling in current awareness and SDI
- Data base mining for case-based reasoning and information fusion

Moderately Successful

Depending on the cleverness of the encoding scheme, the domain knowledge incorporated, and the size of the data set, an application of neural networks might be successful in:

- Interface design
- Retrieval generalisation
- Adaptive retrieval (learning from the past)

Not Successful at All

Due to various problems of scalability the following applications are not expected to be very successful.

- Speeding up traditional systems by parallel computation
- Clustering for bibliographic information and thesaurus generation
- Storing a data base or index in a neural network

In the next chapter we will asses the utility of these intuitions on a large body of reported applications of ANN's in IR as found in the literature.

5 Existing Applications of ANN's in IR

In this chapter the state-of-the-art in neural network applications in information retrieval is reviewed. As many as possible existing cases from the literature are presented, categorised by application type. Some are discussed in more detail than others.[4] *In all cases, the impact of the use of neural networks for the particular applications is emphasised.*

5.1 Serials and Loan Management

Some research towards the application of neural networks in serial- and loan management has been done in the past. However, work presented in this field is rare. The main reason being that loan- and serial management is a task that requires structured and accurate information. As mentioned before, these are features a neural network does not have. The only property a neural net could have in this application is the prediction of non-linear time-series such as generalisations about loan behaviour.

The most important work has been presented in [Hauser et al., 1993]. In this work a pilot project is presented that involves automatic document delivery in response to computerised interlibrary loan requests. Each document request includes an unstructured comment field that patrons occasionally use to indicate whether or not they want the National Library of Medicine (NLM) to fill that request. These comments vary widely in content, but were found to always contain the text 'NLM'. They describe a technique to automatically reduce the amount of operator intervention to resolve ambiguities in the intent of the patron as to whether the request should be filled or not. Nothing is known on future extensions of the pilot.

[4] Some of the reports presented here are based on information from abstracts from the DIALOGUE on-line database. This was only the case for that part of the references of which no full copy could be obtained. In all other cases, a more thorough description is given, based on the original literature.

5.2 Clustering

Applications of information clustering can be found in various papers. The main problems in clustering applications are caused by the fact that clustering is a very expensive process and that neural networks cannot easily cope with the enormous data amounts that are processed in information retrieval applications. In almost all cases, the models work fine with small data sets, but collapse as larger sets are applied.

A detailed motivation of neural networks for clustering in information retrieval can be found in [Ginsberg, 1993].

A Dynamic Thesaurus and its Application to Associated Information Retrieval

In [Kimoto et al, 1991, 1993] an information retrieval system based on a dynamic thesaurus was developed utilising the connectionist approach. The dynamic thesaurus consists of nodes, which represent each term of a thesaurus, and links, which represent the connections between nodes. Term information that is automatically extracted from a user's relevant documents is used to change node weights and generate links. Node weights and links reflect a user's particular interest. According to the authors, a document retrieval experiment was conducted in which both a high recall rate and a high precision rate were achieved. This is a very localist model, that is mainly interesting for spreading activation properties.

Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval

[Chen et al., 1993] show a blackboard-based document management system that uses a neural network spreading-activation algorithm which lets users traverse multiple thesauri as discussed. Guided by heuristics, the algorithm activates related terms in the thesauri and converges of the most pertinent concepts. The system provides two control modes: a browsing module and an activation module that determine the sequence of operations. With the browsing module, users have full control over which knowledge sources to browse and what terms to select. The system's query formation, the retrieving, ranking and selection of documents, and thesaurus activation are described. Here too, a very localist approach is chosen, thereby limiting the properties of the neural net. This is more an advanced implementation of a semantical network than a "real" neural network application.

Automatic Recognition of Semantic Relations in Text

In [Liddy et al., 1991], the goal of the project is the automatic extraction of knowledge from a machine-readable dictionary. Longman's Dictionary of Contemporary English, with its restricted defining vocabulary, provides an appropriate corpus for the development of relation-revealing formulae (RRF), which couple a semantic relation to predictable linguistic patterns in definitions. The maximum coincidence search (MCS) technique is used in the task of delineating these RRF by detecting and retrieving all instances of reoccurring patterns of adjacent and non-adjacent word combinations. The templates which are developed from this data are then used to train the neural network text analyser (NNeTA), whose network topology is modelled on the conceptual organisation of Roget's International Thesaurus (1962). The theoretical aim of the project is to demonstrate the feasibility of combining the symbolic and neural-network approaches to semantic processing of text.

This is a localist approach, in which the author hand structures the entire network. However, the interconnection weights are derived automatically using a statistical algorithm.

Semantic Networks and Associative Databases

In [Lim et al., 1992] two models, one originating from an artificial-intelligence paradigm and the other from database research that incorporate connectionist techniques into their knowledge representation and reasoning processes are described. The first approach, called evidential reasoning, is based on semantic networks and focuses on solving inheritance and recognition queries using a rich internal structure. The second approach, called the associative relational database, provides a query language to manipulate knowledge stored in simple uniform structures. In addition to solving ordinary information retrieval, associative databases support robust retrieval with imprecise queries, which is impossible in traditional databases. The two modelling techniques are compared.

Here too, the emphasis is more on data storage and retrieval than on real "information retrieval" problems such as they occur in libraries.

Simulation of Search Term Generation in Information Retrieval by Propagation in a Connectionist Lexical Net

In order to retrieve information from a bibliographic database the searcher has to translate a natural language problem description into an expression of a query language that can be processed by a retrieval system. This requires a careful selection of the search terms which is normally done by a human searcher. The aim of this project was to simulate this process on a computer. Based on a connectionist approach, a lexical net was built using fully computerised procedures. The associative processes that are involved in human memory when making lexical decisions were simulated by the propagation of activities in the net. [Rapp et al., 1990].

The model that has been used in this research has a localist data representation and is mainly used as disambiguation model.

ART 1 and Pattern Clustering

In [Moore, 1988] an overview of clustering information with an ART 1 neural network is given. The patterns that are clustered could be "information retrieval" type patterns. The model is very small and scalability issues are not discussed. No extensions of the models are known.

More ART-1 Networks and Information Retrieval

More on the application of an ART-1 network for information retrieval can be found in [Caudell et al., 1991], [Caudell 1992a-b]. Although much of the work is implemented with limited data sets, the application of searching engineering designs (textual and graphical) is interesting.

Artificial neural networks have been applied to engineering design retrieval. ART-1 networks are used to adaptively group together similar engineering or graphical designs. The information used to group the parts is coded into binary representations which, in their basic form, amount to bit maps of design descriptors. This technology has been used to build neural databases for the retrieval of two- and three-dimensional engineering designs. The authors discuss a feasibility-level system that learns to group sheet metal parts for modern airliners into similar families, and then to recall the best matching family when presented with a new design.

An addition to the algorithmic form of ART-1 was introduced that allows it to operate directly on runtime encoded vectors, and to generate compressed memory templates. According to the author, the performance of this compressed algorithm, compared to the regular uncompressed algorithm on real engineering designs, demonstrated significant savings in storage of the input vector and the memory templates.

Neural Architectures for Clustering in Document Databases

In [MacLeod, 1990a-b], and in [MacLeod et al., 1991] the suitability of current neural models in performing document clustering is examined. ART-1 as well as Back Propagation models are examined. His papers describe a neural model which has been designed specifically to perform document clustering by feature extraction, using unsupervised learning and tuned system parameters and reports on experimental results obtained from the clustering and subsequent querying of a well known document collection. These results are compared to those obtained by Griffiths and Willett (Report to British Library R&D Dept. on project SI/G/564, 1984) using several agglomerative clustering algorithms.

A difficulty in implementing document clustering using algorithms based on sequential architectures is that in the classification of documents eventually a computational bottleneck arises. Neural networks have the potential to alleviate this problem. The MacLeod algorithm, a neural network algorithm designed specifically for document clustering, is presented. The features of this algorithm are examined and experimental results from two small test collections reported. Based on these results the algorithm exhibits effectiveness comparable to hierarchic (sequential) clustering algorithms. The MacLeod algorithm also appears to require time and space complexities of O(n/sup 2/) and O(n), respectively. Experimental results show that the algorithm's performance is order independent.

Comparable results have been obtained by [Fritzke et al., 1991] in the application of the Travelling Salesman Problem.

Information Retrieval in Sparse Associative Memories

In [Ceccarelli et al., 1992] the authors consider a sparse associative memory with vanishing connections. They prove that it gets a logarithmic storage capacity and an optimal storage efficiency. It is also shown how and how much the storage capacity increases for highly correlated patterns and a sparse input coding. Inferential properties of this model are also investigated. In this paper, the emphasis is more on general information storage and retrieval than on textual information storage and retrieval problems.

Clustering Documents with a Simple Recurrent Network

In [Wermter, 1991] one of the first truly distributed clustering algorithms can be found.

The author describes a recurrent connectionist model which learns to classify book titles from a library. This task poses several difficult constraints to a simple recurrent network (SRN): learning sequences of words, detecting the context of preceding words, assigning a class, and dealing with variable length, syntax, and semantics of available titles. The author describes his underlying word representation, the connectionist model, and experiments with titles from an on-line library classification. The model learned to classify almost perfectly in comparison with the existing library classification. This research shows that a recurrent connectionist model can learn the necessary knowledge for scaling up to 'real-world' title classifications in natural language processing.

Stefan Wermter classifies documents into categories. Wermter uses an unsupervised model that is capable of deriving new categories on the spot.

A simple recurrent network (SRN) is used. In order to avoid the combinatorial explosion of the needed number of neurones, only the words in the titles are used to categorise the documents.

In a way, this work is closely related to the work done by Elman himself on the unsupervised categorisation of linguistic objects. Instead of simple sentences, Wermter used titles and a clever encoding scheme [Wermter, 1991].

Here too, the restricted word set and the use of titles only limit the model's capabilities. Due to the explosive growth of the neural network, scalability appears to be quite impossible.

Clustering Documents with a Self-Organising Feature Map

Another clustering effort based on a fully distributed data representation scheme can be found in the work of Xia Lin. Kohonen feature maps are known to cluster related objects (given some features) into related categories. In information retrieval, documents can be clustered in related groups, so retrieval of associated documents can be facilitated.

In [Lin et al., 1991], the authors train a Kohonen feature map with a number of vectors derived from a set of scientific documents. A predefined set of words is selected. The relative frequency of each of these words is represented by one dimension in the feature vector. Next, depending on the words used in the document titles, a feature vector representing the document can be obtained. These vectors are used in the training phase.

Because titles holding the same words probably discuss the same subject, they should be represented by neurones in neighbouring regions, yielding a common subject (see Figure 5.1).

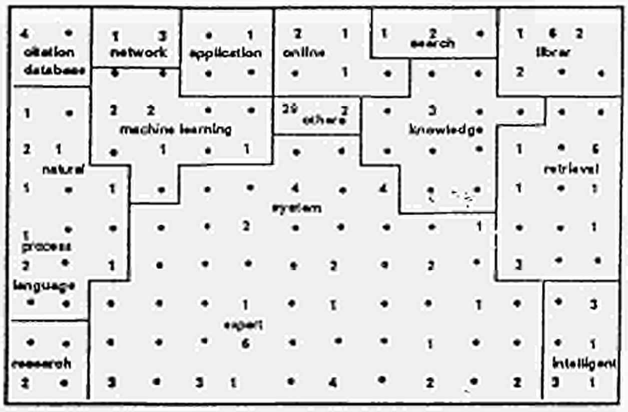


FIGURE 5.1: A SELF-ORGANISING SEMANTIC MAP (REPRINTED FROM [LIN ET AL., 1991]).

This work too has a number of unsolved questions:

- Clustering is known to be expensive, in particular in dynamical environments. This holds also for Kohonen feature maps, which demand longer training times than straightforward statistical techniques.

- Kohonen feature maps can cluster small amounts of data. In the case of large data collections, the feature maps can entangle easily.

- The data is clustered on a preselected number of words occurring in the title only. The model probably collapses if one uses abstract-only or full-text clustering. Moreover, as almost anywhere else in IR, only global surface properties of the titles are used, no structure or meaning whatsoever is involved.

Clustering Documents with a Self-Organising Neural Net: The Neural Interest Map

In [Scholtes, 1993], statistical properties (n-gram or keyword distributions) from various texts are taught to a fully distributed feature map. A comparison of a query with this feature map results in the selection of texts closely related to each other with respect to their contents.

The neural interest map derives a full-text mapping from the documents in the data base onto a self-organising feature map. Related documents are stored in neighbouring areas in the neural network. A query is formulated by the user and matched against the data in the neural network. All documents represented by neurones within a certain distance from the best matching unit are retrieved and presented to the user.

Assume a full-text data base and a limited vocabulary in a specific domain (about 1,000 words). Each text in the data base can be represented by a vector holding a dimension for the frequency of every keyword. By teaching the keyword vector for every text to the Kohonen feature map, a topological representation of various interests will occur. Such a map might be seen as a neural interest map, where related papers are clustered in adjacent neighbourhoods. The main difference between this method and work done by [Lin et al., 1991] is that this model uses the full text (or that of an abstract) to cluster the papers, where Lin only uses 25 keywords occurring in paper titles. The number of keywords used here is much larger (500).

The map formed might be seen as a semantic map of the data base texts. Since [Doyle, 1961] there has been research towards the automatic formation of such maps. Doyle expressed his desire to use the computer not only as a tool in searching, but as a method of discovering semantical relations. This approach is quite similar to the neural network formalism of Kohonen. [Ritter et al., 1989b] and [Ritter et al., 1990] show possible applications of such self-organising semantopical maps in the derivation of semantic relations between words. Moreover, there is literature on the functional specifications of a user friendly interface for document relations [Crough, 1986]. The specifications pointed out here strongly resemble the characteristics of the Kohonen feature maps. Although the relation between the cognitive and semantic maps as meant in the literature and the Kohonen formalism is not that direct, the Kohonen feature maps do share some properties with cognitive maps. Kohonen maps express relations between objects in Euclidean distances, and they are able to reduce complex relations in an n-dimensional feature space into a lower two- (or three-) dimensional space with conservation of spatial and topological relations. More on research towards the cognitive map can be found in [Lakoff, 1988], [Regier, 1988], [Chrisley, 1990], and [Palakal et al., 1991]

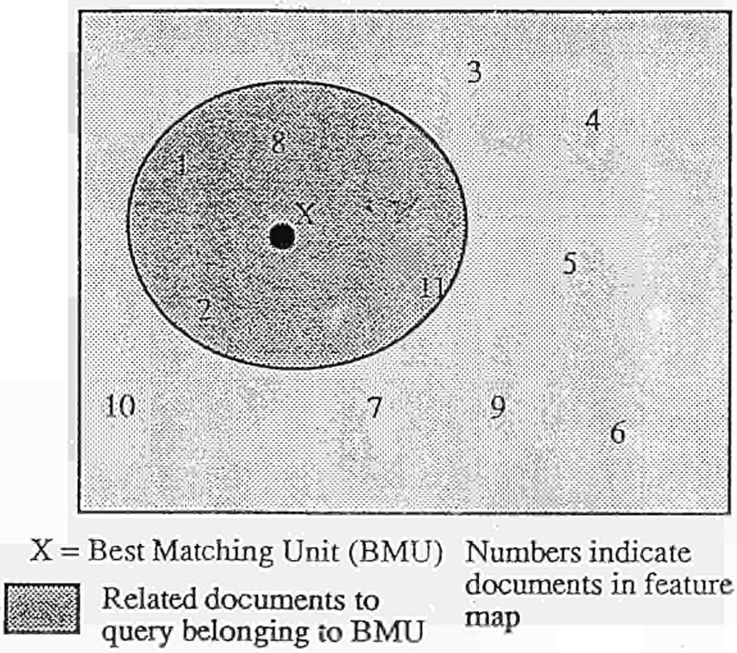

3
4
8
1
X
5
11
2
10
7
9
6
X = Best Matching Unit (BMU)
▨ Related documents to query belonging to BMU
Numbers indicate documents in feature map

FIGURE 5.2: THE NEURAL INTEREST MAP PRINCIPLE(REPRINTED FROM: [SCHOLTES , 1993]).

Simulations and Results Neural Interest Map

A vector representing text distribution features can be derived for every paper in a data base. Such a vector then represents a fingerprint of a data base object. Fifty papers were scanned and their corresponding vectors were taught to the neural network. The Kohonen training mechanism was just standard. No special features were used. Two aspects should be pointed out before the simulations are discussed. First, these simulations differed from the ones proposed by [Gersho et al., 1990a-b], [Wermter, 1991] and [Lin et al., 1991] in that the former used very restricted text parts for the derivation of the feature vectors (mostly titles) and that the methods were based on a well optimised hand-made keyword selection. In the model as presented here, the keywords were derived from the text in the objects automatically and the feature vectors are based on keyword distributions in the training text. In other words, the model as presented here implements a full-text model. Therefore, the vectors have very high dimensions and require long training times. The result is a fully automatic clustering mechanism.

Results

The following feature map was obtained:

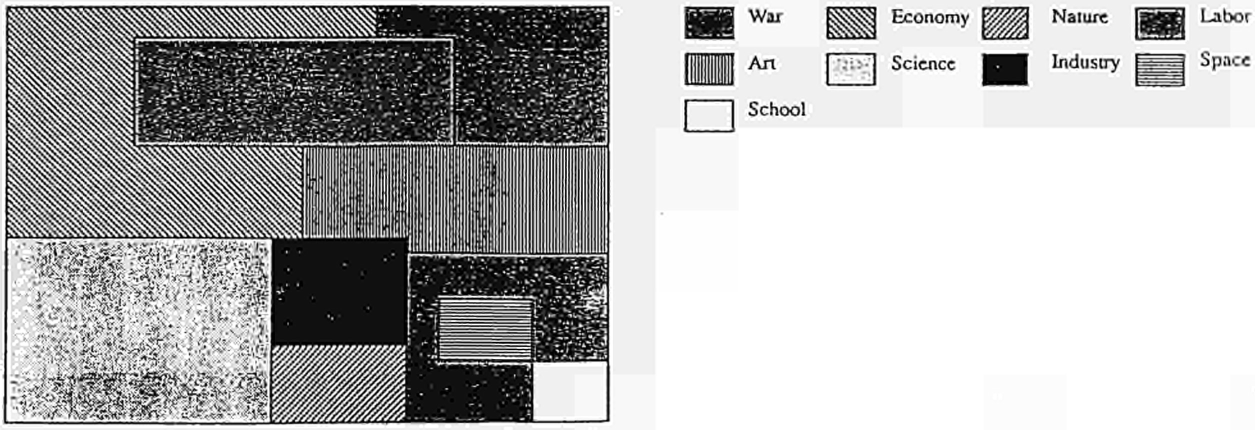


FIGURE 5.3: THE FEATURE MAP AFTER ORGANISATION

Although some groups are still separated for reasons which are not yet clear, the overall impression is that this map holds many of the semantical relations between the documents. However, much of the relations are not correct. The training times required were very long and it is doubted whether the model works on larger data sets. More on the limitations of feature maps can be found in the chapter: "Discussion".

5.3 Interface Design

Neural networks have often been used as spreading activation and generalisation models in interface design. The exact relation with a neural net can be doubted, as most models are hand-made and do not implement "real" neural features other than the integration of multiple sources of information.

It is the question if one really needs a neural net in many of the cases that shall be presented here. In almost all cases, a "semantical network" is constructed for a certain information retrieval domain (legal, medical). Unless otherwise mentioned, these models use localist data structures and are hand-constructed.

Other advanced (non-neural) models are described in [Salton et al., 1994], [Sparck Jones, 1991], and [Savoy, 1992]. Here one can find comparable (non-neural) techniques to the ones presented here.

Motivations for neural networks in interface design and information disclosure can be found in [Osborn, 1992], [Raan et al., 1993], [Morch, 1992].

Information Retrieval as an Interactive Activation Model

Credited for the first application of neural networks in information retrieval is Michael Mozer. He was particularly interested in the IR systems used in bibliographic searches.

The model consists of two sets: a set of descriptors and a set of documents. The descriptors are in fact index terms that can be derived from the documents automatically. As the model is used, the user query activates the descriptor units which in turn activate the relevant

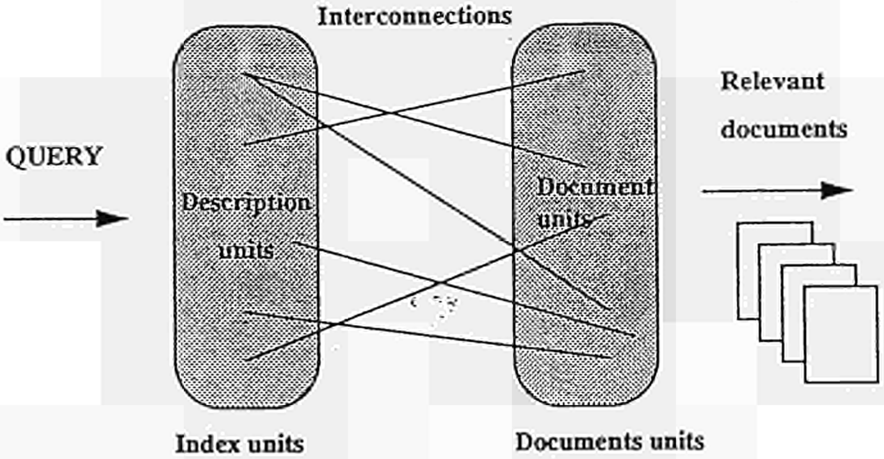


FIGURE 5.4: MOZER'S INTERACTIVE ACTIVATION MODEL IN INFORMATION RETRIEVAL.

The neural activation level is used to indicate the document's relevance ranking. One can doubt the relation with a neural network, however, the model shows some interesting properties caused by the spreading activation [Mozer, 1984]

The AIR System

The AIR (Adaptive Information Retrieval) system was the result of a Ph.D. study by Rik Belew. The model has the same structure as that of Mozer: documents and terms are represented as nodes and associations between them are represented as weighted connections. Whenever a query was set to the index terms, the related (or connected) document units would be activated. Almost all localist information retrieval models presented in this chapter are based on, or derived from this model.

New to Belew's model is the ability to learn (through a localised reinforcement rule). His learning algorithm has three main properties:

- The addition of new documents in the data base results in the automatic adjustment of the connections between index terms and document terms because the total sum of the outgoing weights is kept constant. As a result, more frequent index terms yield a lower activation level.

- The model is well aware of contextual relations due to the adaptive character of document structures. Words frequently occurring in each others neighbourhood will increase each others activation level.

- Users can indicate whether they like or dislike the retrieved documents. As a result, the weights between the nodes is adapted. This highly interactive browsing method is a very efficient relevance feed-back method.

Belew's work is considered to be one of the most comprehensive in early connectionist information retrieval, in particular because he succeeded in positioning neural networks in IR in a very successful way [Belew, 1986, 1987, 1989][Belew et al., 1988]

Other comparable early models of connectionist IR models can be found in [Personnaz et al., 1986], [Cohen et al., 1987], [Bein et al., 1988], CRUCS [Brachman et al., 1988], and ZZENN [Cochet et al., 1988].

SCALIR, a Hybrid Symbolic & Connectionist Models in Legal Information Retrieval

Belew's AIR did not use any real semantical (sub symbolic) information. As a result, retrieval results lack any deep structure. In SCALIR (Symbolic and Connectionist Approach to Legal Information Retrieval), Rik Belew and Daniel Rose extended the AIR model with a semantical network.

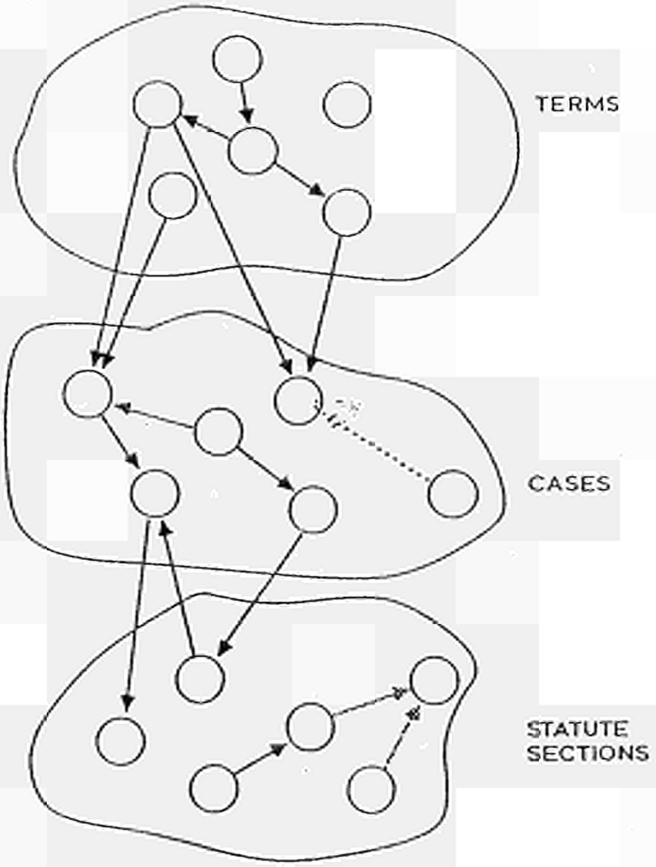


FIGURE 5.5: THE SCALIR ARCHITECTURE. NODES ARE REPRESENTED BY CIRCLES, SOLID LINES ARE CONNECTIONIST LINKS, DOTTED OR SHADED ARROWS ARE SYMBOLIC LINKS (REPRINTED FROM [ROSE ET AL., 1991]).

A clever activation function combines the proper connectionist and symbolic links. The connectionist links are called C-Links (these are the same links as used in the AIR system), the symbolic links are called S-Links. Units can be connected by either C-Links or S-Links. Both links participate in the activation value of the unit (see Figure 5.6, "The S-links and the C-links used in SCALIR")

Training of the network connections is done with a similar rule as in the AIR model, however, instead of keeping the total of all outgoing connections constant, a an algorithm similar to the delta-rule is used to adopt the weights.

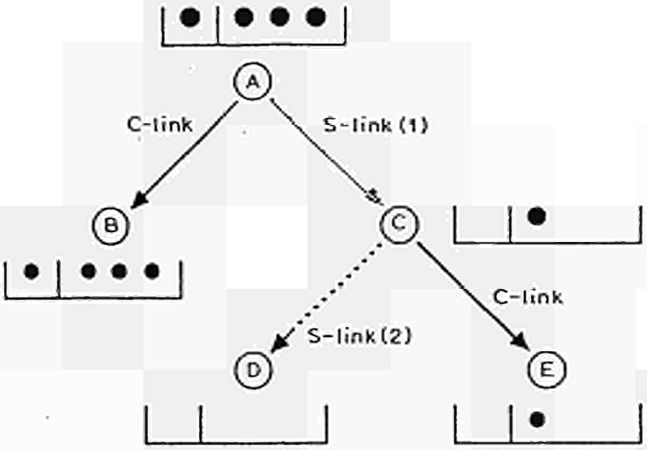


FIGURE 5.6: THE S-LINKS AND THE C-LINKS USED IN SCALIR (REPRINTED FROM [ROSE ET AL., 1991]).

As with the individual work of Rose, the model does not add much to the neural sciences. However, it is very successful in mapping the notions used in the information retrieval to those of the neural network community [Rose, 1990, 1991], [Rose et al., 1989, 1991].

Associative Representation of Concepts in Neuronal Networks

[Wilbert, 1991] shows a realisation of an associative representation and access on semantically specified concepts within concept lattices by neural networks. The use of a feature-oriented concept analysis is essential for the described realisation. If documents will be indicated and retrieved by such a concept representation, there succeeds a solution of the synonymy and homonymy problems as well as an improved content-oriented access.

First it takes a critical look at existing approaches of associative retrieval and spreading activation techniques since the models uses only these properties of a neural network in this localist model.

Virtual Text and New Habits of Mind

In [Carlson, 1991], a motivation for neural network application in hypertext in information retrieval is given. According to the author, hypertext is just one of several forms which electronic publication can take. At the same time, it is the one which most emphatically demonstrates that, unlike print, there need not be a match between the physical and the logical structure of a piece of writing. Because they are so totally divested of the conventions of print, hypermedia systems can take on many forms. Based on this versatility, the heart of hypertext (chunks and links) can seamlessly be combined with three areas emerging in software development today: intelligent applications, neural networks, and scientific visualisation, to create new discovery tools for the mind.

Much has been written on hypertext systems and neural networks. The nodes of a localist neural network are often used to implement hyperlinks. As the reader may understand, this has little to do with neural networks.

A Connectionist System to Assist Navigation in Hyperdocuments

In [Biennier et al., 1990 a, b, c] it is shown that browsing through a hyperdocument or a document base is often hard for the reader who does not easily reach the information he searches. The author's retrieval method uses the structure of the information base and a neural thesaurus. This system relies not only upon a semantic description of the user's needs, in terms of multimedia keywords or tags, but also upon the indication of the precision and specialisation levels desired by the user for the required information. A neural network connects tag-cells (in a thesaurus of multimedia key words) with the hyperdocument nodes. Nodes are selected according to their level of activation in the neural network. An inertia parameter governs context migrations, and a filter restricts context expansion. The activation process interferes with the hypergraph of the hyperdocument. All these parameters are involved in the cell's activation rule in order to provide a permanent adaptation to the user. A browsing path, which gives a view of the base adapted to the user's needs is then dynamically built.

In [Biennier et al., 1992] it is shown that integrated manufacturing and engineering involve a system able to manage large and heterogeneous data and knowledge, especially for concurrent engineering and one of a kind production: text, CAD graphics, technical data, manufacturing information. Traditional knowledge bases are designed in a fixed way. Objectives and knowledge are detailed and are not taken into account in a global way. If new technologies such as hypertext systems seem to be convenient to store this unorganised information and knowledge, they are not sufficient to manage it in a dynamic way. That is why the authors couple the information base to an epigenetic neural network. External events or users needs are used to activate this network which retrieves and organises all the needed information. By creating, adapting or deleting neurones and connections, the knowledge structure evolves gradually and is adapted to the users needs. The system has five main interests: the knowledge base evolves dynamically during all its life; the system is able to react towards external events by structuring its information in a convenient way; the system can also co-operate with other neural networks which are used to pre-process the information; as they build it in an incremental way, the system can rely on hypertext systems to store the information base; and it can easily be coupled with information retrieval systems.

Nevertheless, all the models presented by the authors are based on a localist data representation scheme.

Russian Hypertext

The authors present a neural network that is designed to work with hypertext systems in applications involving the manipulation of large quantities of text. Use is made of a network in which text units are linked to significant words found in them, the weight of links being determined through an automatic text analysis based on a normalised word frequency measure. There is no intermediate layer, making it possible to construct the initial state of the network rapidly, and to readily accommodate new documents as they arrive over a period of time: a peculiar requirement of the application. Emphasis is paid to the method in which the user can provide feedback based on the value that the user attaches to the documents retrieved in response to a query [Gedeon et al., 1991].

Good relationships are Pivotal in Nuclear Data Bases

The authors expound the importance of effective use of information in the nuclear industry. As a result, they received several requests for a full article on this subject. In this article, they start with the tenet that valuable information is stored in nuclear experience data bases that must be capitalised on for enhanced operation of plants, training, and rule-makings. After an introduction, a method of adaptive information retrieval based on neural network methodology is introduced, followed by an example [Heger et al., 1991].

Spreading Activation Methods in Information Retrieval- a Connectionist Approach

An improvement of spreading activation methods in information retrieval (IR) systems is designed in this article. An appropriate type of neural network was developed as a new tool. The neural network, called RETRIEVALNET, is a composition of two sub-nets (recursive and feed-forward) extended by a layer of inhibitory neurones [Husek et al., 1992].

The Effects of a Dynamic Word Network on Information Retrieval

[Iwadera et al., 1992] describes a method of learning a user's field of interest and the effects of applying this method to information retrieval. This method uses a Dynamic Word Network (DWN) within the framework of an Associated Information Retrieval Approach. The Associated Information Retrieval approach aims at retrieving easily and precisely the information that a user needs out of a database. To achieve this, the information retrieval system must understand what the user intends to retrieve, that is, the user's interest. An associated Information Retrieval System (AIRS) that incorporates this approach is now being developed. AIRS learns the user's interest from sample documents and represents the user model as a DWN. A DWN consists of nodes and links. Each node corresponds to a term which AIRS can use for retrieval and each link corresponds to the relationship between two terms. Each node also has a node weight. To evaluate DWN performance, we retrieved information using AIRS comparing the output with conventional methods. The results show how the DWN improves the precision of information retrieval.

The Adaptive Network Library Interface

The evolution of the concept of an adaptive network library interface is described and several technical and research issues are explored. The Adaptive Network Library Interface (ANLI) is a computer program that stands as a buffer between users of the library catalogue and the catalogue itself. This buffer unit maintains its own network of pointers from book to book, which it interactively elicits from the users. It is hoped that such a buffer increases the value of the catalogue for the users and provides librarians with new and useful information about the books in the collection. The relation to concepts such as hypertext and neural networks is explored as well [Kantor, 1993].

A HyperNet Approach to Literary Scholarship

Connectionism can offer the literary scholar and student a method of indexing hypertext documents in ways which can uncover patterns of similarities among text segments that might otherwise not be noticed. The basic idea of connectionist networks (also called neural networks) is explained and this idea is then applied to the analysis, or mapping, of texts. Finally, the basic idea of mapping texts with connectionist networks is incorporated into a design for a Macintosh computer application, called HyperNet, which is explained in detail. HyperNet is meant to provide the scholar, teacher, or student with the opportunity to easily configure and reconfigure a hypernet mapping of whatever textual data he or she wants to subject to a close analysis of textual feature resonance [Koch, 1992].

A Neural Network Integrated with Hypertext for Legal Document Assembly

Hypertext technology is increasingly finding application in law firms, since much of the work of a lawyer involves accessing, assessing and sculpting text. There are some severe limitations inherent in hypertext. In document assembly, users break documents into small units (clauses and subclauses) and link them into a complex web, but links between text units do not cater for all associations of ideas that users may wish to make. Much research aims to provide means of accessing information not restricted by the hypertext structure. Recent methods include free text retrieval (FTR), vector retrieval. FTR is unsuitable for complex legal applications-it is necessary to distinguish between clauses which may share a common vocabulary. Vector retrieval does not allow indirect associations of words, documents. The authors have implemented a novel, neural network-based approach to information retrieval in legal hypertext systems. Practical considerations in the design include handling an often changing collection of text units [Mital, 1991].

A neural network designed for retrieving legal information is presented. Studies have shown that conventional full text retrieval of legal documents gives an extremely poor recall. It is possible to mitigate the problems by adding subject labels to documents, but these labels soon either become too general or proliferate in such quantities that users find it difficult to remember or find the relevant ones. The problem arises from the facts that: legal concepts are open-textured and cannot be readily classified, and a large number of concepts are expressed using a small number of technical terms. The authors tackle the problem by using an interactive network with two layers of bi-directional links between units representing documents and words respectively. Some of these links are set according to a normalised, relative word frequency measure obtained through an automatic analysis of the text. Other links are used during learning. The architecture is designed partic ularly to cope with the addition of new documents which would change the existing relative word frequencies without degradation in performance [Mital et al., 199x].

An Adaptive Document Retrieval System Using a Neural Network

Current document retrieval systems based on keyword retrieval have many problems in regard to human-computer interaction. The paper focuses on three of them. Firstly, not only the inputted keywords but also their relations carry important information as a representation of the query, though the system ignores the relations and uses them as indices of documents. Secondly, as a word can be understood in various ways by different people, one query can represent various sets of keywords. Thirdly a user who does not have enough knowledge about his target field may fail to represent his query as a set of the keywords. A document retrieval system that overcomes these drawbacks is presented [Mori et al., 1990].

Current document retrieval systems assume that a user can represent his/her query using natural language or keywords. The user without much understanding about the subject being searched for often cannot even phrase the query properly. The authors propose an interactive document retrieval system to overcome this human-computer interface difficulty. Their system modifies its responses according to the user's evolving mental state. The system makes inferences about the users' subject knowledge based on their response to information it presents. This also allows the users' requests to be more quickly focused and effective. The system proposed in this article is implemented by using the neural network technique, and the user's query is represented by the activation pattern over the localist neural network [Mori et al., 1991].

Integration of a Connectionist Model in Information Retrieval Systems

[Mothe et al., 1992] proposes the use of a two-layered neural network to implement a connectionist approach in information retrieval and to perform an automatic query reformulation using user's dissatisfaction feedback.

After an outline of information retrieval, the author proposes the use of a formal neural network to implement associative information retrieval mechanisms. She shows graphic comparisons of the results obtained from experiments in which parameters underwent variation and shows how this connectionist model query permits better performances than classical information retrieval systems through automatic query reformulation [Mothe, 1992].

Cluster Analysis, Graphs, and Branching Processes

This article presents (1) a general formalism for cluster analysis, allowing a systematic study of simulation research, in particular its dynamic aspects, (2) a model of small bibliographical clusters, allowing inference (among others) on the connectivity of domains, and (3) an outline of new theories of networks with randomly changing nodes and edges, applicable for analysis of different types of relations, e.g., communication between scientists, etc. These models may be useful for analysis of large databases in artificial intelligence. They may also have significance as new approaches to neural network analysis [Nowakowska, 1990].

MNEMOSYNE, a testbed for ANN's in Information Retrieval

MNEMOSYNE, a testbed for the comparison of different neural network architectures and learning algorithms, has been created to demonstrate potential applications of neural networks in the field of information retrieval. The pattern clustering algorithm of [Klassen and Pao, 1989] is incorporated into MNEMOSYNE. By encoding lexicon terms into sparse numeric matrices which become the inputs to the clustering module, it is possible to produce clusters of lexically related terms which are generally also semantically related. The testbed also supports the formation of fuzzy cognitive maps (FCM) [Kosko, 1988] which enable the retrieval of sets of terms related to a given vector of input terms by a form of constrained spreading activation. Sets of retrieved terms may be stored for future reference in a temporal associative memory (TAM). The FCM is initially hardwired, but is able to learn in real-time by competitive differential Hebbian learning. The authors give a discussion of how FCM's with competitive differential Hebbian learning and adaptive resonance theory might also be employed to support hypertext [Oakes et al., 1993].

PThomas

[Oddy et al., 1991] reports the state of development of PThomas, a network based document retrieval system implemented on a massively parallel fine-grained computer, the Connection Machine. The program is written in C, an enhancement of the C programming language which exploits the parallelism of the Connection Machine. The system is based on Oddy's original Thomas program, which was highly parallel in concept, and makes use of the Connection Machine's single instruction multiple data (SIMD) processing capabilities. After an introduction of systems like Thomas, and their relationship to spreading activation and neural network models, the current state of PThomas is described, including details about the network representation and the parallel operations that are executed during a typical PThomas session.

KNOWBOT

The adaptive interface KNOWBOT was designed to solve some of the problems that face the users of large centralised data bases. The interface applies the neural network approach to information retrieval from a data base. The data base is a subset of the Nuclear Plant Reliability Data System. The interface KNOWBOT pre-empts an existing data base interface and works in conjunction with it. By design, KNOWBOT starts as a tabula rasa but acquires knowledge through its interactions with the user and the data base. The interface uses its gained knowledge to personalise the data base retrieval process and to induce new queries. The interface also forgets the information that is no longer needed by the user. These self-organising features of the interface reduce the scope of the data base to the subsets that are highly relevant to the user needs. A proof-of-principal version of this interface has been implemented in Common LISP on a Texas Instruments Explorer I workstation. Experiments with KNOWBOT have been successful in demonstrating the robustness of the model especially with induction and self-organisation. This paper describes the design of KNOWBOT and presents some of the experimental results [Sharif et al., 1991].

Parallel Associative Processes in Information Retrieval

[Wettler et al., 1990] presents an autoassociative net which describes the associative connections between 269 words in a scientific reference database. This net is used to predict which words professional searchers will use when generating queries on the basis of written problem descriptions. These predictions are compared with on-line searches conducted in natural settings.

Incorporating the Vector Space Model in a Neural Network Used for Document Retrieval

This research describes investigations in implementing a document retrieval system based on a very limited neural network model.

The model is very localist and it has very little to do with a neural net. Everything is hand-constructed and not even features such as lateral inhibition and spreading activation are used. [Wilkinson et al., 1992].

A Neural Network Approach for User Modelling

A neural network approach to user modelling is proposed in the context of information retrieval. User level of expertise and the inquiry interests underlying the goals are considered as two major factors affecting information provision. A prototype system, UM-net, is presented for modelling user domain experiences the inquiry interests and tailoring the descriptions about software packages provided to a user [Chen et al., 1991].

A research framework for building a user model system by utilising artificial neural network (ANN) approaches is proposed. First, some problems in user modelling are discussed which underlie the motivations of introducing ANN approaches. Second, some considerations concerning ANN properties and their applications in task-related user modelling are presented. Finally, an ANN-based, integrated user modelling system is proposed which incorporates conventional symbolic reasoning approaches in a multilevel processing environment [Chen, 1992].

An Adaptive Information Retrieval System Based on Neural Networks

Partial results of an experimental investigation concerning the use of Neural Networks in associative adaptive Information Retrieval are presented. The learning and generalisation capabilities of the Backpropagation learning procedure are used to build up and employ application domain knowledge in the form of a sub-symbolic knowledge representation. The knowledge is acquired from examples of queries and relevant documents of the collection. In the paper the architecture of the system is presented and the results of the experimentation are briefly reported [Crestani, 1993].

Learning Query-Documents Set to a Back-propagation Network

In the well known work of K.L. Kwok, a fully distributed back-propagation network is used in probabilistic information retrieval by extending it with adaptive edge growing capability, equivalent to query expansion. That is, the model is adaptive with respect to the results of a query. Thereby the model is capable of showing adaptive and generalising capabilities.

As mentioned before, information retrieval can be seen as the mapping from queries to relevant documents. K.L. Kwok collected such large sets of query-documents pairs from the use of a large IR-system. The query as well as the documents were represented by vectors in which every dimension represented the weighted occurrence of an index term.

Next, he trained these vector combinations to a back-propagation neural network. As a result, the model is capable to generalise unknown queries to known document (sets) [Kwok, 1989, 1990, 1991a, 1991b].

This model really added some value to connectionist IR systems. First, Kwok uses his neural network as a generaliser for unknown queries. He is not interested in clustering or category derivation, but only in document retrieval. By doing so, he implements an interesting mapping in a back-propagation network. In fact, if one should use this network in addition to known techniques such as inverted indexes, relevance ranking and relevance feed-back, very interesting new properties can be observed. However, the number of possible query-document pairs grows exponentially with the size of the data base. Therefore, it is difficult to scale to larger systems.

Categorising Documents with a Back-propagation Network

Often, information retrieval has been presented as the mapping from queries to relevant documents. A simplified case of this mapping is the mapping of documents in predefined categories. Whenever documents are categorised into categories, retrieval is expected to be easier and faster. In general, this categorisation is quite time consuming and therefore an expensive, manual process. Because the back-propagation network is known to be a real achiever in such non-linear mapping problems, it has been applied several times.

Documents can be represented by inverted indexes, semantical concept vectors and other representation schemes. The n-gram representation is known to be robust and easy to use. David Mitzman and Rita Giovannini used the n-gram document representation to train a back-propagation neural network the mapping between documents and categories.

Over 5,000 document vectors were manually classified into all possible categories. Every document was represented by the frequencies of the bigram vector (all occurring two letter combinations). The training set consisted of structured [bigram vector, category] pairs. After training the model was capable to classify the test set of 25,000 documents 95% of the cases into the proper categories [Mitzman et al., 1990].

The main problem in this approach is the difficult scalability of the bigram vectors to e.g. trigrams (the addressing space grows exponentially, resulting in neural networks of unacceptable sizes). In addition, although n-grams are known to be fast and robust, they provide no more than a simple surface analysis. Due to the supervised nature of the model, the automatic derivation of new categories is not possible (which was no problem in the application the models was used for).

Fault Tolerant Hashing and Information Retrieval Using Back Propagation

The architecture and performance of neural networks designed and trained to computer hashing functions is described. The networks described are of the connectionist type and are capable of learning complex mappings using the back-propagation or error algorithm. Connectionist networks are robust, are capable of limited error correction, and offer several advantages over traditional hashing methods. Multiple indexing, which implements many-to-one mapping, can be easily realised by training a network for each key attribute. The neural network approach can be used to train a very large number of pattern associations by dividing a problem into smaller problems. This neural network consists of several sub-networks, each solving a specific mapping task. The experimental results show that small neural networks with simple processing elements can learn complex mapping that implement index search in constant time [Dontas et al., 1990].

HNC's MatchPlus system

HNC is developing a neural network related approach to document retrieval called MatchPlus. Goals of this approach include high precision/recall performance, ease of use, incorporation of machine learning algorithms, and sensitivity to similarity of use. The implementation of MatchPlus is motivated by neural networks, and designed to interface with neural network learning algorithms. High-dimensional vectors, called context vectors, represent word stems, documents, and queries in the same vector space. This representation permits one type of neural network learning algorithm to generate stem context vectors that are sensitive to similarity of use, and a more standard neural network algorithm to perform routing and automatic query modification based upon user feedback.

In fact, the vector space relevance ranking algorithm is extended with a neural component, implementing interesting semantic relations. Especially the combination of traditional and neural methods is very interesting. [Gallant et al., 1992].

In the TREC-1 conference, this system structurally underperformed with respect to traditional vector space models. However, several adaptations" in the TREC-2 conference that make the system " less neural show a model that does outperform traditional techniques. At this very moment. HNC is marketing the system as a commercial product.

Document Retrieval Using a Neural Network

A key requirement of large organisations is to manage large volumes of natural language text. One might wish to locate relevant documents from a collection of one million documents. The task is to match one natural language query against a large number of natural language documents. Neural networks are known to be good pattern matchers. The paper reports the authors first investigations in implementing a document retrieval system based on a localist neural network model [Hingston et al., 1990].

Fuzzy Cognitive Mapping of On-line Search Strategies

Artificial Neural Systems (ANS) offer one means to both model and evaluate search strategies employed in on-line database searching. Using an ANS application, a sampling of search strategies employed in an academic setting and in a special library setting were examined. The ANS model employed was a fuzzy cognitive map, as introduced by Bart Kosko, (FCM) which allows both supervised and unsupervised learning settings. Both of these settings are present in on-line database searching models. The FCM was chosen because it does not exhibit stable point behaviour, but does exhibit oscillatory or limit behaviour, approximating the search process in the on-line environment. The FCM model was employed on two sets of actual on-line search strategies, one from an academic library, the other from a special library [Hurt, 1991].

This model is based on a distributed data representation scheme that resembles the ART-1 and Kohonen feature map.

Neural / Query Search Software

In [Shaw, 1993] a program called Neural / Query is described that searches data bases and builds networks of partial matches and confidence factors. With a concept dictionary, the program provides best guess answers. The derivation of the network is based on a statistical algorithm. This is the only known commercial application in this area.

Perceptrons in Information Retrieval

In [Wong et al., 1991-1993] a method is given for computing term associations based on an adaptive bilinear retrieval model. Such a model can be implemented by using a three-layer feed-forward neural network. Term associations are modelled by weighted links connecting different neurones, and are derived by the perceptron learning algorithm without the need for introducing any ad hoc parameters. The preliminary results indicate the usefulness of neural networks in the design of adaptive information retrieval systems. Because the model uses linear activation functions and a limited training algorithm, its ability is limited.

CONET-IR

CONET-IR is a connectionist network for intelligent information retrieval. It provides a network architecture and a methodology for reference query negotiation in a neural network environment. This negotiation serves as a primer to information retrieval from an information system. A conceptual graph is used to represent queries. Concepts in a query are separated from their relations by an attentional connectionist network. Concepts, relations, attributes, entities and functions are stored in a knowledge base. The analogical reasoning mechanism will select typical queries similar to the ones entered by the user, so the user might fill in the blanks for the concepts, relations, attributes, functions and entities on the fly.

5.4 Filtering of Information

Introduction

The usage of neural networks as adaptive user-interest-models has been tested by a number of researchers. Here the results are much better than in cases described in the clustering and user-interface sections. The main result is the natural relation between the application and the neural networks. In addition, user interest models have to store limited amounts of information, making them more suited for the application of the current limited neural networks.

Current Awareness and Selective Dissemination of Information (SDI)

In current awareness and selective dissemination of information (SDI) systems, large amounts of information are monitored, filtered or ordered with respect to some user-interest model. The task of these systems is to select the most relevant information with respect to some user model. In the past, these user models were defined according to some Boolean query. One can understand that these models are not adaptive what-so-ever and that they are therefore hard to maintain.

As the amount of information that organisations have to process is increasing, more and more interest developed towards current awareness and SDI applications. Some of the workshops and conferences that spend attention to this subject were:

- [BANKAI, 1991] where adaptive information filtering in financial environments was an important topic,
- [Bellcore, 1991], a general workshop in (multi-media) information filtering,
- [EUROMICRO, 1992], where the subject was discussed in a ESPRIT context, and
- [Wall Street, 1991] where one dealt with topics such as news understanding and information filters.

More general background articles and the relation to the information highway can be found in [Newquist, 1994] and in work carried out by Negroponte from MIT Medialab on personal news services and "The Daily Me".

Commercial products for SDI and current awareness are: Digital's experimental project *Mailfiler*, Verity's *Topic Real-Time* and ZyLAB's *ZyFILTER*. Personalised news services can be obtained through Internet, Compuserve and other computer networks. The best known service that uses full-text selection algorithms is *Heads-Up*. All these programs use standard Boolean search techniques and are non-adaptive.

Knowledge Representation in Information Retrieval with a Simple Recurrent Network & User Modelling by Using a Simple Recurrent Network.

All commercial SDI and current awareness models use global surface analyses only (keywords, Boolean relations, proximity relation, etc.). The relation between semantical networks, information retrieval and neural networks could well be observed in the local connectionist IR models. Due to the lack of structural representation schemes, semantic networks were not applied in fully-distributed neural IR models until Bob Allen used a simple recurrent network (SRN) for such an application.

The addition of semantics is considered essential for the retrieval results. However, there are two main problems. First, they have to be added by hand. Second, this often slows down the system to an unacceptable rate.

Allen trained a SRN with pairs of question and answers. The questions were composed of multiple entities such as "who-is mother-of mary". Answers were single objects like "sue". A possible training sequence would then be: "who-is mother-of mary * sue", where * indicates the separation between a question and an answer. Over 32 input terms, 27 output terms and, 8 persons were composed to a corpus of 626 sentences. After training the model was capable to answer various simple questions [Allen, 1991].

Allen hoped to apply his findings in a information retrieval system as a module to add highly structured semantical information to flat textual surface properties. The scalability of this method may be doubted for large data bases. This is partly due to the complex relations obtained as the data base grows and partly due to the explosive growth of neurones in the SRN as the questions become more complex and larger in number.

A Personal News Service

Early work in the field of user-modelling can be found in [Jennings et al., 1992a,b, 1993a,b]

Here, some new methods for accessing very large information services are presented. The authors propose the use of a user model neural network to allow better access to a news service. The network is constructed on the basis of articles read, and articles marked as rejected. Over time it adapts to better represent the user's interests and rank the articles supplied by the news service. Using an augmented keyword search one can also search for articles using keywords in conjunction with the user model neural network. Trials of the system in a Usenet news environment show promising results for the use of this approach in information retrieval.

The model is completely hand-build and is based on a localist data representation scheme. Therefore, it is not as adaptive as one would expect and it definitely does not implement typical features such as generalisation and association.

Expert Assistance for Collection Development

Collection development may be seen as an extension of SDI and current awareness. Therefore this reference is listed in this chapter. [Johnston et al., 1990] discusses a project to develop an expert system to assist with the selection of material for libraries. After a consideration of the suitability of this domain for expert system development, possible forms of knowledge representation are presented, including the form used in the prototype. The types of knowledge which a selection expert may have are outlined, and there is some discussion of deep knowledge in relation to the brittleness of many existing expert systems. Part of the project involves building a learning component into the system, and two approaches, induction and neural networks, are outlined.

In some later works, the authors discuss a machine learning approach to knowledge acquisition in the domain of monograph selection. Collected data collected was used as input for an induction program and for an artificial neural net. The rules generated by the induction algorithm are analysed, and compared with the expert's assessment of the value of the criteria used. The generated rules and the localist neural nets were tested and both were found to perform tolerably well. The results indicate that machine learning could have an important part to play in the development of expert systems in this and other domains which involve the allocation of scarce resources [Johnston et al., 1991].

A Neural Network Approach to Text Processing

According to [Sunthankar, 1992], the text search problem can be broken down into two main tasks; database searching and message routing. Database searching consists of searching through a large database of text from certain key words, phrases or other simple functions of strings. Message routing is classifying incoming messages and sending them to the appropriate 'mail box'. The author discusses and compares current leading edge solutions to this problem and introduces some new ideas based on recent neural network theories and experiments. All text-search and retrieval technology is predicted on the assumption that the semantic content of text can be predicted from its syntactic properties. The classification of messages is done by using a localist neural network.

Text Classification by a Neural Network

When banks process their free-form telex traffic, the first task is the classification of the telexes. Historically, several attempts have been made to automate this process, using various stock phrases as features, on which to base the classification. This poses a problem because there are large amounts of data available, but the rules for classification are not explicitly available. For solving these kinds of problems, neural networks have the advantage of extracting the underlying relationships between the input data and the output classes automatically. Based on this consideration, the authors have built a neural network classification system, which has three subsystems: a user-maintainable feature definition subsystem, a feature extraction subsystem, and a localist neural network subsystem. The neural network is simulated on a VAX computer with a fast learning algorithm, and is combined with some non-statistical knowledge from the feature definition system. Above 90% correct recognition rates have been achieved for the major categories concerned [Wei, 1991].

Filtering the Pravda with a Self-Organising Neural Net

[Scholtes, 1993] contains a chapter that presents an implemented neural methods for free-text data base search. A specific interest (or "query") is taught to a Kohonen feature map. Next, large amounts of unstructured text are passed along the network. Depending on the activity patterns that occur on the network, a text can be selected by the system.

Various simulations show that for both networks, the neural network indeed converges towards a proper representation of the objects that are taught. The algorithm seems well scaleable (linear time and memory complexity), resulting in high speeds, few memory needs, and easy maintainability.

The neural filter implements a mechanism in which a (large) query or interest description, stated in natural language, is taught to a self-organising neural network, which derives an internal representation of the text. This text is then matched against a large stream of incoming data. In short, the query is stored in a feature map, the data base is matched against this query. In a practical implementation of this model, multiple queries can be matched simultaneously (see Query A to D).

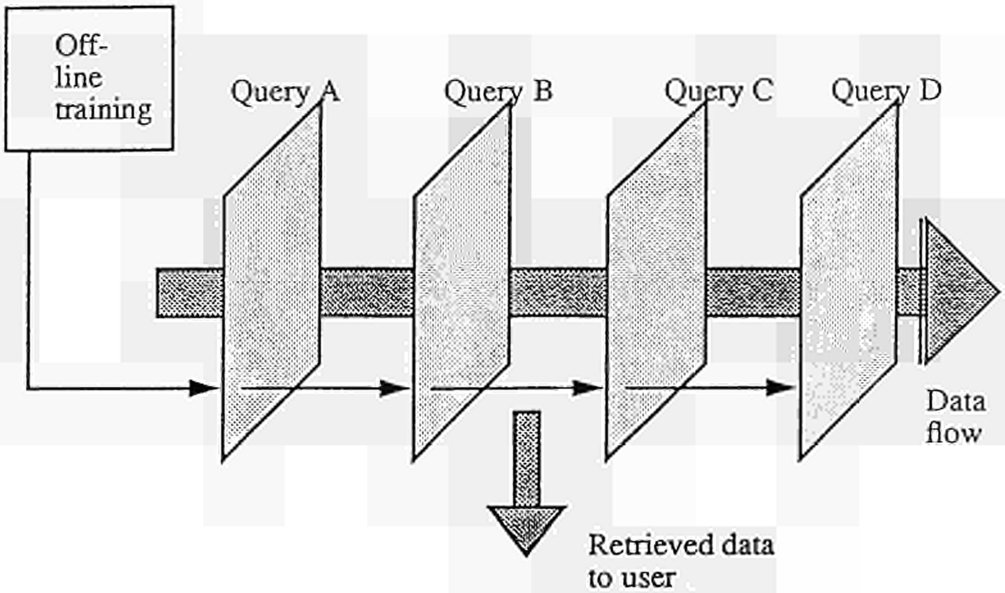


FIGURE 5.7: THE NEURAL FILTER PRINCIPLE (REPRINTED FROM [SCHOLTES, 1993]).

The model is implemented in a Kohonen Feature Map by using statistics about the adjacency of elements in the underlying text.

A statistical algorithm that incorporates such adjacency information is the n-gram vector method. An n-gram is an n-length sequence of characters occurring in a word. For example, the trigrams (n = 3) occurring in the word trigram are --t, -tr, tri, rig, igr, gra, ram, am-, m-- (the - indicates a space). An n-gram frequency vector (which is equivalent to an nth order Markov chain over characters) can be viewed as a document finger print; documents can be identified by such vectors. Normally, bigrams (2-grams) are not distinguishing enough, trigrams (3-grams) yield enough distinction and can be practically calculated, 4-grams do not add enough difference in feature vectors to justify the computational power, 5-grams are almost impossible to calculate and resemble keyword vectors. N-gram vectors provide enough distinguishing power only if common words and common endings are eliminated from the text trained to the neural map. Furthermore, by multiplying n-gram frequencies with weight values (high values for rare n-grams and low values for frequent n-grams), less frequent n-grams may be accentuated.

This model has some major drawbacks.

- First, there is the Markovian nature of the model. It cannot remember strings longer than the order of the Markov chain, even when a larger context is relevant to distinguishing two objects. One can extend the order of the chain, but every step results in a rapidly increasing search space. So, the n-gram method is not really scaleable to higher order dependencies (e.g., 5-gram character or word chains).

- Second, the implementation of higher order n-grams requires sophisticated programming techniques. The statistical tables must be hashed, ordered, and normalised.
- Third, there is no meaning involved in the analysis method; only structural features of the text are taken into account.

Nevertheless, n-gram vectors are very powerful, easy to manipulate, easy trainable and language independent [Forney, 1973], [Hanson et al., 1974], [Neuhoff, 1975], [Shinghal et al., 1979a-b], [Hull et al., 1982], [Srihari et al., 1983, 1985], [D'Amore et al., 1988], [Kimbrell, 1988].

The Kohonen Neural Filter Based on Characters

The n-gram analysis method can be interpreted as a window size n, shifting over the words. This can be implemented quite simply in the Kohonen input sensors by assigning several sensors to each element in the window and concatenating all the window sensors to one big input vector. By shifting this window over the training text, only frequent n-grams form clusters on the feature map, the others are overruled.

After training, texts corresponding best to the query in the feature map will fit best to the clusters in the map (i.e., will yield the lowest cumulative error with respect to the input values). Thus, this type of feature map can be used as a filtering device in an environment with a static query and a dynamic information flow. The method can be extended by incorporating spaces, so the model also learns adjacency relations between words. Such n-gram frequencies are often correlated with syntactic, semantic and sometimes even pragmatic information.

The neural filter based on words

In the second neural filtering algorithm, the system has access to a small dictionary of 500 to 1,000 words. Every word has an unique code of some sensor values. After elimination of non-relevant words (words that are not in the lookup table) and word-endings, a vector representing a Markov chain over words is calculated. This vector is taught to the system. After passing the training text multiple times, the Kohonen feature map contains a representation of common word combinations in the training text (see figure 4).

By processing the retrieval text similarly, the retrieval algorithm incorporates contextual relations. The measure of correlation between these vectors and the representation on the feature map, determines whether a text part can be selected or not. In this example all words are taught to the network. However, sometimes a word does not occur in the dictionary (because it is irrelevant for the selection process). The model ignores these words. As a result, it determines context from the relations between the remaining words.

Training and Retrieval Rules

The training rule used in the previous model is the Kohonen rule. It does not matter whether one uses characters, words or large n-grams as input elements, a coding procedure prepares all symbolic data for input to the feature map by translating it to vectors. This coding process is performed with the aid of a lookup table. All elements of the training set are assigned randomly to specific codes in this lookup table. The codes themselves are spread homogeneously through the feature space, to speed up the training process. Convergence parameters as proposed by [Ritter et al., 1989a] fine tune the Kohonen rule.

Once the feature map training is completed, one must match the test data with the representation formed on the neural map. In the case of the neural filter, one counts the cumulative (normalised) error or the cumulative (normalised) number of perfect hits (or some variant of these two functions).

In general, one can separate two types of selection rules: positive and negative ones. The negative approach mainly filters the noise. A more positive approach is to choose possible candidates for selection. Negative selections are mostly normalised, while positive ones are not. If one paragraph in a paper is related to a specific interest, the positive filter selects it directly, where the negative one ignores the one paragraph due to normalisation of the retrieval value (one paragraph fires high, all the others low, so the average firing level is still low). Positive selection mostly results in too many candidates where negative selection results in too few candidates. A proper combination of both approaches results in the best retrieval results.

Possible positive search methods are plain keyword matches and the (non-normalised) number of perfect hits on the neural map (in the case of n-gram on characters as well as n-gram on words). A negative filter is the added and normalised error of all text elements with respect to a statistical table or a neural map.

Simulations and Results Neural Filter

The training set holds a small selection on the 1987 nuclear weapon restriction talks between the USA and the USSR. The test set was the entire Pravda CD-ROM, being passed along the neural filter.

> "Our era, a fast-paced era of nuclear weapons, an era of growing economic and political interdependence, precludes the possibility of security for one nation at the expense of others. I repeat: we can only survive or perish together. Security today can only be viewed as mutual, or to be more precise, universal. So whether we like each other or not, we need to learn how to coexist and live in peace on this small and very fragile planet. Question: Do you support the continuation in 1987 of the Geneva talks between Soviet and American representatives for the purpose of achieving progress on the issue of limiting and reducing arms? Answer: Yes, we do. We support talks that would overcome the state of fruitlessness and inertness and acquire true dynamism, in a word, talks that would become genuine talks on reducing nuclear arms and preventing an arms race in space. We tried to achieve that in Reykjavik and will try to achieve it even more energetically in 1987. I am sure that such a radical turnaround in the talks would respond to the vital interests of the American people as well. At the same time, the position of the US administration on this issue is a cause of great disappointment for us. After Reykjavik the American delegation in Geneva has become even less co-operative. Despite the fact that the USSR has not been conducting nuclear detonations for 18 months, the USA has continued tests and refused to discuss a total ban on them, though it committed itself to conduct negotiations on that issue in the two treaties of 1963 and 1974. In November that was aggravated by the provocative action the White House took when it broke the important strategic arms limitation agreement (SALT II). It does not help to conduct successful negotiations on new agreements when the old ones are being deliberately and blatantly broken. This is a serious problem that deserves very close attention. I will state once again that we support agreements on the most radical reductions of arms, both nuclear and conventional. Now it is up to Washington."

TABLE 5.1: TRAINING SET (OR 'QUERY').

Error Measurements in the Training Process

By measuring the error during the training process: $||wr(t) - x(t)||$, an insight in the convergence properties of the neural network can be obtained. First, one has to understand that this neural network is used as a selection- and ordering device. Due to a smaller size than needed, only the most frequent n-grams are remembered (or trained properly), all others are forgotten, or overruled. Therefore, the average error will remain high (due to non frequent n-grams). In the first graph the total error in time is plotted. Globally, the error tends to decrease with the number of training cycles. The high errors on the right are infrequent trigrams that must be forgotten. (These are errors of n-grams which are continuously being bounced out).

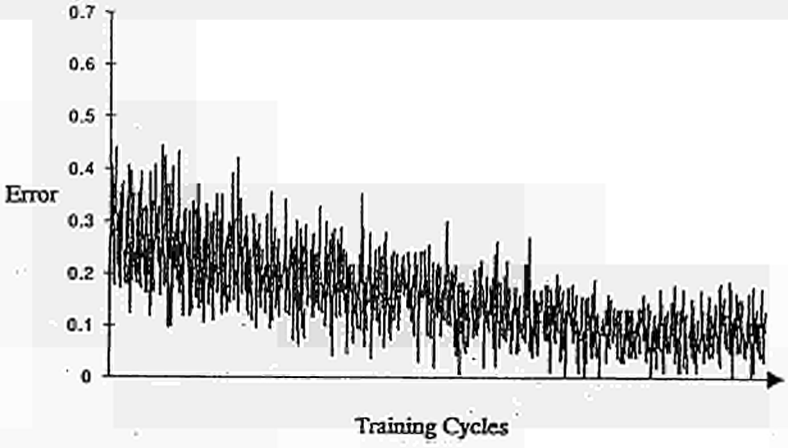


FIGURE 5.8: ERROR DURING TRAINING PROCESS.

The next graph plots the error if it is smaller than 0.05. By plotting these bars, one sees that the frequency of almost perfect hits increases in time: the density of bars is much higher at the right side of the graph than at the left. This indicates that the model is getting better at representing n-grams.

Retrieval Results of the Neural Filter

The selection quality is a measure that cannot be given without be.ıg partial. In Chapter 2, the notions precision and recall are explained in more detail. It was argued that these values are very subjective and only relevant if compared to other techniques for the same data base and the same queries. However, even then, one might question the reliability of these numbers. Nevertheless, some precision and recall values are calculated here in order to make a comparison possible.

The precision and recall distribution, given some parameter for one of the most advanced statistical Information Retrieval techniques [Croft et al., 1991], can be found in Figure 5.9 below.

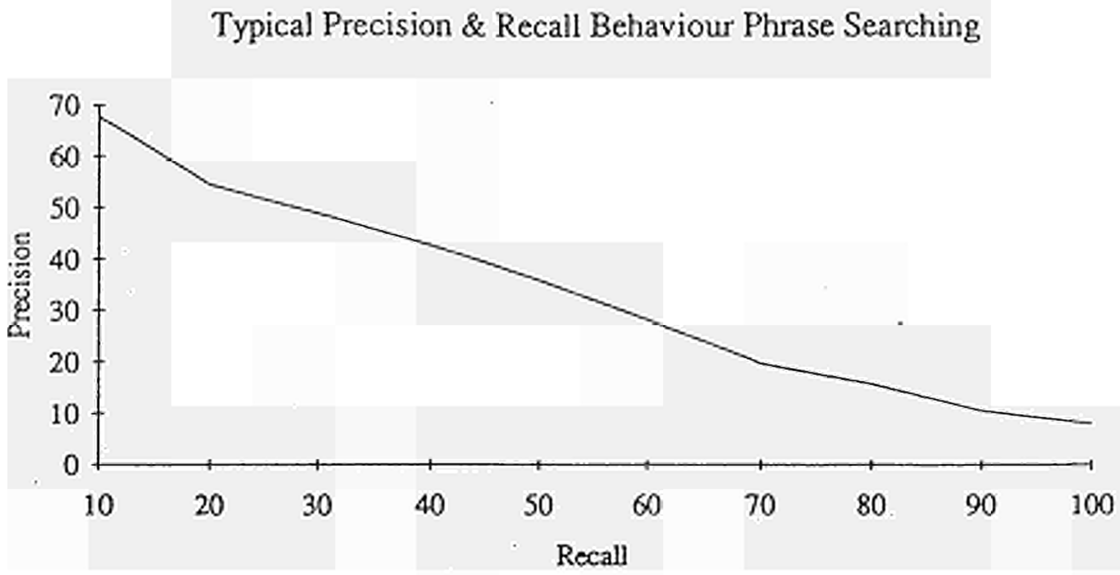


FIGURE 5.9: TYPICAL PRECISION AND RECALL DISTRIBUTION FOR A PHRASE SEARCH IN A STATISTICAL INFORMATION RETRIEVAL SYSTEM.

One can clearly see that both precision and recall are never higher than 45%. If one of the two is set higher (by some parameter change), the other decreases dramatically.

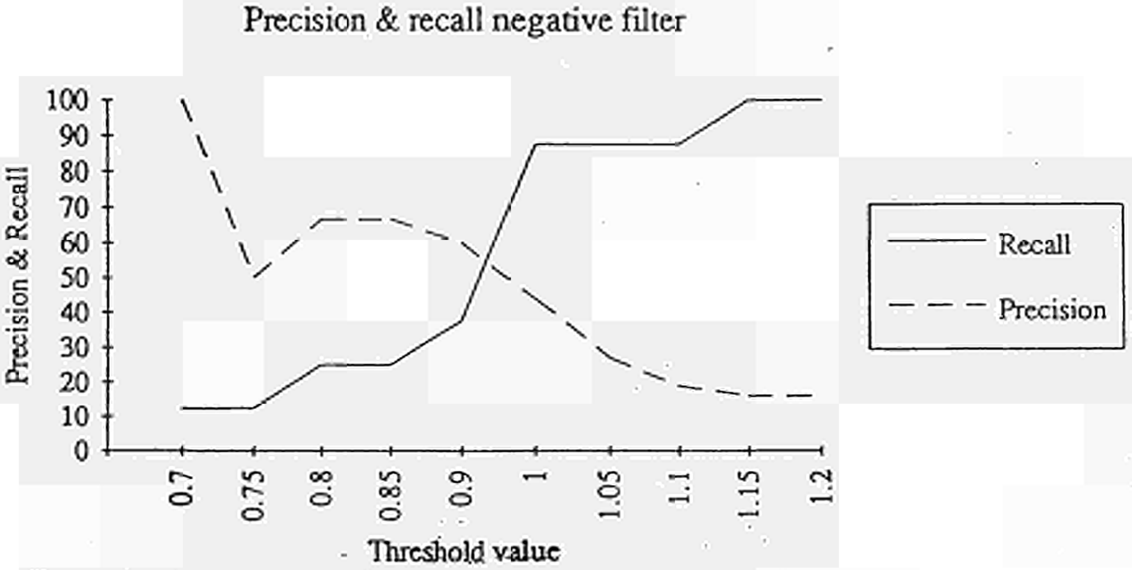


FIGURE 5.10: PRECISION AND RECALL FOR THE NEURAL FILTER

One can clearly see that precision and recall are much higher for the neural filter than they are for traditional statistical IR techniques.

In order to incorporate more context, a neural net is used to derive an internal representation of a certain interest. As it is argued, this self-organising neural net derives a map of conditional probabilities that are somewhere in between simple Markovian windows and Shannon information theoretical values. By doing so, the filter incorporates more contextual information in the filtering process than models that are based on adjacent n-gram analysis, all this without loss of speed.

In this project, the Kohonen feature map is used as an associative memory for the user interest model. In the process of mapping the large stream of information to this neural network, multiple retrieval algorithm can be used. The determination of the most efficient retrieval function is a domain for study in itself. Obvious experiments can be done about combining a negative and positive training rule. More mathematically based correlation functions can be incorporated, etc. This is a main topic of future research. Pointers can be found in the literature on statistical pattern recognition [Sammon, 1969], [Duda et al., 1973], [Small et al., 1974], [Fu, 1977], [Croft, 1977, 1980, 1981], [Bokhari, 1981], [Devijver et al., 1982], [Voorhees, 1985], [Siedlecki et al., 1988].

5.5 Incomplete Searching

Introduction

The commercially most successful application of neural networks in information retrieval is searching in incomplete data-sets such as data gathered by an Optical Character Recognition (OCR) program. Although OCR technology is getting better and better, even the best programs make errors that need to be traced down for proper retrieval capabilities. In general, the costs for cleaning-up OCR text varies from $ 2 up to $ 5 per page. A good overview of these typical OCR problems can be found in [Rice et al., 1992], [Robertson et al., 1993], [Stoddard, 1992], [Sun et al., 1991], [Taghva et al., 1993], and [Wu et al., 1991].

Often, this type of error matching is referred to as fuzzy retrieval, although it has nothing to do with "fuzzy logic". For the sake of clarity, here we shall only refer to fuzzy retrieval if the program really uses fuzzy logic. Otherwise it will be referred to as "searching in incomplete data sets". Some motivations for the use of fuzzy retrieval and neural network for incomplete matches can be found in [Costello, 1992], [Bordogna et al., 1992], [Nordell, 1991].

Programs like Excalibur and ZyIMAGE store the scanned image (a bitmap) in combination with the OCR text. A "fuzzy" retrieval algorithm matches a query to the data set, thereby taking "typical" OCR errors in account. As a result, the text can be located and retrieved. By popping-up the original image, the "real" text can be studied (including graphs, tables, logos, formulas and signatures).

Excalibur produces a number of text-imaging products that search through large amounts of text, video and images using binary pattern recognition algorithms. It uses neural networks to find "related" patterns in text, images and video (images in time). References to Excalibur can be found in [Anthes, 1993], [Blanchard, 1992], [Busch, 1992], [Computer Letters, 1993], [McCormick, 1992], [Myers-Tierney, 1992], [Schwartz, 1993], and [Simpson, 1993].

ZyIMAGE uses an algorithm based on a template search on the index. The templates that are used are generated by an algorithm optimised for OCR errors: replacements, insertions and deletions. This program does not use neural network technology, but it does a very good job.

Another product using neural networks for incomplete searching is Info Select, a Personal Information Manager (PIM) for Windows. The program combines various information such as telephone, addresses, fax numbers, etc. It uses advanced search methods based on neural net technology to locate fuzzy information, misspellings, phonetic variants, etc. More references on this product are: [Emigh, 1992] and [Gilliland, 1993].

Information Retrieval Using Hybrid Multi-layer Neural Networks.

In [Gersho, 1990a-b], it is shown that a hierarchical hybrid neural network comprising simple neural networks provided significantly higher accuracy in data retrieval than single neural network architectures. Both approaches were applied to information retrieval from large databases using textual retrieval keys where either the retrieval key or the data in the database are noisy. The results were improved by using different network training methods for highly correlated and less correlated data. The combination of self-organising and supervised learning neural networks solved this problem, providing a retrieval accuracy of 93% when presented with noisy data, providing a fast training time, and allowing the solution to be scaled up.

Associative Dialogue System (ASDIS)

The aim of the ASDIS system is to process natural language input statements into a data base retrieval language. The neural component is mainly used to disambiguate and perform incomplete matching. A generation of natural language answers is aimed for. Until now, a word recognition system has been developed with flexible association that can put words into correct order in spite of typing errors and can also associate terms with similar meaning through the use of a knowledge base. The grade of flexibility (error tolerance or similarity) can thereby be changed interactively. However, the model is based on a localist representation scheme, limiting the trainability, and generalisation power of the program considerably [Deffner et al., 1990a-b][Deffner et al., 1991].

Conventional and Associative Memory-based Spelling Checkers

In [Cherkassky et al., 1990, 1992] conventional and emerging neural approaches to fault-tolerant data retrieval when the input keyword and/or database itself may contain noise (errors) are reviewed. Spelling checking is used as a primary example to illustrate various approaches and to contrast the difference between conventional (algorithmic) techniques and research methods based on neural associative memories. Recent research on associative spelling checkers is summarised and some original results are presented. It is concluded that most neural models do not provide a viable solution for robust data retrieval due to saturation and scaling problems. However, a combination of conventional and neural approaches is shown to have excellent error correction rates and low computational costs; hence, it can be a good choice for robust data retrieval in large databases.

Moreover, based on several ad hoc models for associative spelling checkers, a generic model is proposed that incorporates powerful N-gram encoding for word representation and supervised-learning associative memories. Recent research on associative spelling checkers is summarised and some original results are presented. It is concluded that many neural network models do not provide a practically viable solution for robust data retrieval, due to saturation and scaling problems. However, a combination of conventional and neural approaches is shown to have excellent error correction rates and low computational costs.

Fuzzy Logic with Linguistic Quantifiers in Decision Making and Control

[Kacprzyk, 1992] gives an overview of recent applications of fuzzy logic with linguistic quantifiers is presented. First, the author sketches two calculi of linguistically quantified propositions which make it possible to determine the truth of such propositions. He then advocates that such propositions may provide a means for adequate aggregation of partial scores (pieces of evidence), and may lead to new decision making and control models. Multicriteria, multistage (control) and multiperson decision making models based on fuzzy logic with linguistic quantifiers are presented. Finally, applications in database querying, inductive learning and neural networks are mentioned.

Information Retrieval Based on a Neural Unsupervised Extraction of Thematic Fuzzy Clusters

After a review of the family of unsupervised neural algorithms, that the authors designed for deriving a compact and relevant representation of a documentary database, they present a user interface based on these grounds. This representation is on two levels: a global topics map, and local topic axes, ranking both terms and documents. A prototype, running in a Macintosh environment and implementing a basic version of this browser, is then described and discussed (Neurodoc project) [Lelu et al., 1992].

5.6 Searching in Multi-Media

Multi-media data can be labelled by textual descriptors, which can then be used to locate and retrieve the multi-media information. However, it would be much more interesting to search for a multi-media data file by using a multi-media query. Pictures, video and sound are data collections that are always characterised by noise and incompleteness. Therefore, neural networks are natural tools to make multi-media "query/data" comparisons. In [Henseler, 1993], the author characterised neural networks as "pattern crunchers", and that is exactly what they are used for in this kind of applications.

Good overview articles for general pattern recognition can be found in [Eliot, 1992], [Kelly, 1991]. The storage and (associative) retrieval of images with neural networks can also be seen as a form of data compression. Work in this area is given in [Chen et al., 1992],[Stafylopatis et al., 1992]. The best known commercial product that implements this kind of functionality is Excalibur, as discussed above.

Design Retrieval by Fuzzy Neurocomputing

The need to automate parts of a design process has generated a variety of software packages, such as computer-aided design and manufacturing (CAD/CAM), computer-aided engineering (CAE) and others, that have become conventional engineering tools. The authors examine the feasibility of utilising fuzzy associative memory in design retrieval and illustrate the concept by applying it to a sample problem of selecting an appropriate design for a solar heating system. In particular, a technique in which a design can be automatically retrieved based on how well it satisfies the desired functional requirements or input criteria is introduced, and issues regarding the integration of the proposed system with a commercially available CAD package (Auto CAD) are discussed [Bahrami et al., 1992].

This model uses a neural net type of memory that resembles a Kohonen Feature map.

The British Library's Picture Research Projects

In this project a neural network is used to search in pictures. The model is tested on a large scale and shows remarkable good results [Cawkell, 1993]. The neural models used are:

Image Transformation & Retrieval

Image transformation and retrieval using neural networks. The objective of this research was to develop an automatic feature extraction system to provide the basis of a data base retrieval mechanism for images which contain no indexing information. Each image within the data base is coded with respect to these features and matched against a query image according to the degree of similarity between the target and the data base code. A self-organising neural network architecture is used to extract those features which optimise the performance of the code. This work demonstrates how a neural network based system can perform extremely rapid fuzzy matching of images within large data bases (see the entry *Image Transformation* in the Product and Project References Annex to this report for address pointers).

5.7 Data Mining

Neural networks are good at incomplete searching in noisy heterogeneous data sets. By doing so, they show great abilities in generalising and association tasks. One of the applications in which a user wants to search for noisy information (or use an incomplete query) in a data base that stores various information types (numbers, dates, text, structured records, etc.) is the study of data mining: the search for unknown answers on unknown questions.

Applications for data mining can be found in fraud detection, help desk management, business intelligence, and other application in which one does not exactly know what one is looking for. As more and more information is digitally available and more and more information is prepared by other parties (so we do not really know what is in there), the need for data mining gets bigger and bigger. Much related to this application is the study of Case Based Reasoning (CBR), knowledge based systems that solve problems by browsing and developing inferences over old cases (free-text and structured features). The browser as well as the inference engine can be the field of study of a "data mining" neural network [Foster, 1992], [Levine, 1992], [Olsen, 1994].

According to the authors this is one of the top applications of neural networks in the near future. Overview and motivation articles on this subject can be found in [Lee et al., 1993], [Germain, 1992], [Ford, 1989], [Koekebakker, 1991].

Nestor

The best known commercial product for data mining is Nestor. This OEM toolkit of neural-like pattern recognition techniques is the number one standard for hand-written character recognition and data base mining. It is used in almost all successful commercial products such as zip code reading (USA & Europe) and database mining for Eurocard (fraud detection). Exact technology is very proprietary.

SupportMagic for Windows 2.1

SupportMagic from Magic Solutions, Inc. provides help desk and asset management software that handles help desk support, inventory/configuration management, reporting and more. The software logs, assigns, prioritises and tracks support requests via a central database, with automatic call escalation. SupportMagic for Windows 2.1 sends messages via e-mail to and from MagicWin such as automatic notification of clients when a problem is closed to measure help desk results. Statistical Information Retrieval (SIR) finds calls in the database with similar results somewhat like the results from case-based reasoning tools. DDE capability enables users to open a help desk call from within another DDE-compatible application, query MagicWin for status information and then access the data for other projects.

Top of Mind Help Desk for Windows

This program supports integrated call logging and tracking and uses neural nets and fuzzy search to learn automatically on many levels at once. It provides smart routing and forwarding, intelligent picklists, and smart parsing data entry. Moreover, it provides help desk employees with advanced (neural based) retrieval algorithms (mainly for fuzzy search).

Top of Mind advanced Help Desk uses cognitive processing technology that models human thinking to diagnose calls. Through this technology, the system automatically learns and keeps itself updated on a company's environment from its own interactions with logged cases. Advanced features include integration with third party e-mail and asset management packages. The new advanced version includes multiple choice picklists that allow consultants to select multiple symptoms to a particular problem, adding a new dimension to help desk problem diagnosis and resolution. Hypermedia has also been expanded to support more picture formats as well as full motion video and sound.

A Neural Network to Extract Implicit Knowledge from a Nuclear Data Base

A major task in mining large data bases is to extract knowledge that is implicitly rather than explicitly coded into the data base. In this work, a new technique based on neural networks is applied to extract implicit causal knowledge from the sequential coding search system (SCSS). The SCSS is a data base developed for the US Nuclear Regulatory commission to store information about all incidents that occurred in any nuclear plant in the United States since 1980 as reported by the utilities in licensee event reports. The implicit information in this data base is composed of the causal relationships between various incidents. The implicit causal knowledge can be inferred by analysing the patterns found in explicitly stored information about each incident. Since each incident is recorded as a sequence of chronologically ordered occurrences, where an occurrence is a cause/effect relation, a neural network method will be used to extract an implicit causal knowledge expressed as pair wise relationships between causes and effects and to build on this to develop a novel approach to identifying sequences of cause/effect relations [Gacem et al., 1990].

Credit & Credit Card Data Bases

In [Eliot, 1993] a Neural network is used for searching and analysing customer behaviour in credit databases. According to the various information that is gathered, a credit advise is given automatically. The specific task of the neural net is the combining of all the information.

Another application of analysing credit card user behaviour can be found in [Lewinson, 1994] and [Buta, 1994]. This paper describes a company that uses a general neural network tool for analysing customer behaviour and fraud trends. Unfortunately, due to the confidential character of the material, the article is not very detailed.

Neural Computation in Knowledge Based Systems

In [Barthes et al., 1991] the authors try to prove that the internal processing of KBS may be close to neural techniques. They describe the internal representation of the expert knowledge obtained from a compilation of production rules. The production cycle is only information propagation through a graph of cells, until the stabilisation of the network. The learning process is not yet implemented, but the back-propagation technique is currently used for the backward reasoning process. Their research domain concerns artificial intelligence techniques applied in many expert domains such as information retrieval, technology transfer or medical diagnosis. The characteristics of the KBS of Telemac are indicated. This software is an operational multi-agent system used in medical diagnosis.

An Associative Neural Expert System for Information Retrieval

[Desrocques et al., 1991] reports on a type of architecture that is a synthesis between neural network and expert systems. After describing this new architecture from a quite literal example: i.e. splitting a word into syllables through learning processes, the authors thoroughly study a homographic parser and give the major results they have obtained. Next they examine the advantages that can be expected from that new architecture in the context of an information retrieval application, as well as the best way to use it within existing systems. This model is very structured and symbolic and is based on a "tricky" combination of different AI paradigms.

Neural Net Modelling in Diagnosis and Information Systems

Studies on the use of neural networks for (medical) diagnostic tasks and associative (information) recall are delineated. Each study in the group involves both neural net software and conventional computing, with goals varying from the desire to do a comparative analysis to achieving complementary problem solving. The first study involves methods for clustering, classification, decision making. A second study involves a comparison of a neural net model used in associative recall and more conventional database systems developed in Prolog and in Ingres. In a sense, the set of solutions gives a fourth, fifth and sixth generation perspective on diagnostic and information retrieval tasks. The study is briefly overviewed as a contribution to developing notions of simulation environments, as it is a portrait of the kinds of behaviour suitable for simulation systems and their support [Reilly et al., 1990].

University of Central Queensland

A group at the University of Central Queensland conducted a scientific project that investigates the suitability of Neural Networks for identifying situations where expert systems can be applied by construction of hybrid neural net-expert system in domain of serials management in libraries. See the Product Reference Annex for a contact address.

5.8 Juke-box Staging

Global Information Management

The most creative application of neural network in an information retrieval system has been implemented by the company Global Information Management. This company sells an advanced document imaging system, used by various banks in Switzerland, Germany and Austria. It uses a neural network to predict Juke-box staging in a network environment. This is the planning of the schedule by which CD-ROM discs are placed in the optical drive. As a result, images are loaded more efficiently. The model is automatically trained, based on user behaviour. This is a good application of a neural net since it is self-learning and performs very well in predicting these non-linear time-series.

6 Discussion & Conclusions on State-of-the-Art

"Good text managers do more than search and retrieve,
they bring organization to chaos"

-- Raymond Ga Côté

In this chapter, the objectives of the first chapters shall be discussed with respect to the solutions and models presented in the two previous chapters. First, a general overview of the limitations of neural networks is given. Emphasis is on Kohonen feature maps, as this is one of the most promising neural net architectures of the moment. Nevertheless, much of the Kohonen properties hold for other models too. Next, a framework is given that can be used to predict the success of a neural network in an information retrieval application in a library context. Last but not least, the reader is provided with some directions for future research.

In general, the results of the literature study are disappointing. That is, most neural models that are used in information retrieval pretend more than they can actually accomplish. This is especially the case for all the localist clustering, user modelling and interface design models designed so far. By using the term neural network, the authors pretend having adaptivity, generalisation, association and other neural features. In most cases these features are lacking completely or they are implemented by a complementary algorithm.

The cases in which the results were better could be found in the applications of filtering, noisy data retrieval, data mining and multi-media retrieval. Here the character of neural networks fits better to the natural character of the application. Nevertheless, here too, some researchers were able to abuse the term "neural network".

6.1 Extending Traditional Information Retrieval

The application of neural networks in information retrieval in a libraries context is just one of many directions to extend the power of information retrieval systems. Other directions are symbolic Artificial Intelligence (AI) and Pattern Recognition (PR). Here a small overview is given of the relation between the three.

Both traditional Artificial Intelligence (AI) and Neural Networks (NN) have their appeals and their drawbacks. It is not the intention of this project to compare AI with NN because they can hardly be compared objectively. On the one hand, there are tasks in which AI is simply better, such as the representation of hierarchical structures or the implementation of recursion.

On the other hand, many of the problems that occur in traditional AI can be solved better with NN. As it appears, AI and NN are often complementary; if one of the two is successful, the other is probably not. Therefore, comparisons of the two approaches for a specific task are often not fair, because one of them probably suits the application better than the other.

More interesting is the relation of AI in the field of (statistical) Pattern Recognition (PR). This study has traditionally been concerned with the classification of objects through (low level) signal processing. Over time, only statistical techniques appeared to be relevant. Structural pattern recognition (the AI branch of pattern recognition) already failed many years ago. Many in the field of PR argue that neural networks are nothing else than a fancy method of implementing a (not well understood) statistical classification technique.

It is a fact that Hidden Markov Models (HMM) and other statistical re-estimation techniques have much in common with methods used in neural networks. But there remain a number of significant differences between the sequential statistical algorithms and neural network technology:

- Most of the statistical PR models implement supervised function approximators. Self-organising models do exists, but are not as good as they are in neural network research.

- During the training process, self-organising neural networks show very interesting recurrent interactions, which enable the models to derive organisations and classifications that are based on more than just simple adjacent context dependencies. Kohonen feature maps implement such neighbourhood effects.

- Generalisation and association are implicitly present in the neural network models.

Of course, one could always implement this typical neural behaviour in a sequential statistical algorithm, but then, one has to program these features explicitly one by one (which can be quite a difficult task).

On the other hand, there are some major drawbacks of neural networks that currently limit their full application in the real world.

- If one applies neural technology to an application, it either works, or it does not work. It is impossible to patch the model for certain exceptions, which can be done in a sequential algorithm. This aspect is mainly caused by the distributed data representations and the parallel character of neural networks. The first is responsible for the fact that one does not know where to attach the patch (many neurones are representing many concepts). The latter causes a complicated process in which it is hard to say which neurone is responsible for which action (the credit assignment problem).

- In addition, neural technology is not really scalable. Almost all experiments are based on sequential simulations of parallel algorithms. Parallel hardware is rare and very expensive. Moreover, many of the models tend to collapse if they are applied on large problems.

Statistical pattern recognition does have some proper answers to both problems, making it more suited in many real-world applications. In particular if it uses re-estimation techniques such as is the case in speech recognition with HMM.

6.2 Localist Connectionist Models

Connectionist models or neural networks that use a localist or sub-symbolic data representation scheme (that is, one neurone for one concept) are not neural networks in the sense that they can claim properties such as generalisation, association, adaptivity, and robustness.

All models that are based on this technology are either complicated implementations of semantical networks, hypertext structures or other symbolic models. Spreading activation and lateral inhibition as a disambiguation tool are the only interesting processes that can be observed in these model. The term neural network is misplaced for these models, connectionist models would be a better descriptor.

As long as these models have to be constructed by hand (connections as well as weights), they cannot be applied on a large scale in information retrieval problems (the data sets are just to large to maintain manually).

In general, if the model does not have the following properties: self-organising, distributed data representation, massive parallel structure and natural data input, one should be very careful in considering it to be a neural network. Models that do not implement all these features, can in many cases also be implemented in a statistical model with much less effort and with much more efficiency.

6.3 Kohonen Feature Maps

In this work, neural networks are used for different tasks. In the recurrent Kohonen feature maps, they are used for the processing of language. In [Scholtes, 1993] they are used to derive a contextual representation of some query, and in [Scholtes, 1993], [Lin, 1991] they are used to derive an internal representation of the corpus. The first application explicitly implemented language processing, where the latter two used the feature maps as an associative memory. The processing was carried out by an external sequential algorithm.

Kohonen feature maps implement finite state behaviour in the recurrent model, although stability and convergence speed were not among the virtues of the system. As finite state behaviour is not enough for natural language processing and because the representation of hierarchical structures requires large amounts of neurones (there is evidence that the number of neurones grows exponentially with the number of states, [Servan-Schreiber et al., 1991]), the application of such models is limited. As a matter of fact, many of the language processing models, as they exist in the back-propagation community suffer from similar problems.

A neural network behaves much better when it is used as an associative memory of a larger system rather than a processing device. But there are still a number of problems with the feature maps:

- It is difficult to use the models in larger applications. As soon as the feature maps grow larger, they tend to entangle. One of the reasons for this behaviour is the fact that larger maps have less boundary effects (which are very important for the organisation process). In order to avoid this, during the training process one must try to keep the feature maps as much in order as possible. There are a number of methods to do so. First, one can try to combine many smaller maps into a hierarchical system. Secondly, one can use more context sensitive training algorithms such as the Hypermap or the models presented by Kangas. Another option is the use of growing models as introduced by Martinetz and Fritzke. The feature maps of the latter two are always in order because they start with a restricted set of neurones and only add new neurones if they are needed.

- The feature maps exhibit interesting neighbourhood effects. Due to these effects, contextual relations are incorporated in the derivation of the feature map weights. However, it is not completely clear how much context is incorporated in this process. The Kohonen feature map implements "a very complicated process" that cannot easily be described quantitatively. However, some qualitative indications can be given. It can be argued that the better the form of the feature map adopts the form of the underlying probability distribution, the better the model incorporates contextual relations in the derived conditional probabilities. In addition, the more the map is in order during the training process, the better the derived probabilities will be. Further on, it is argued that a lower bound for those values is an adjacent Markovian conditional probability and that an upper bound is a Shannon information theoretical value.

These two points are heavily interrelated. If the maps entangle less, the map is of higher quality. Although this issue is of great interest to the community, none of the experts in Kohonen feature maps can provide quantitative measurements on the quality of feature maps. Moreover, much effort is invested in better models that are more scalable. If work in this direction is not successful, the Kohonen feature maps may encounter tough times.

Knowledge Representation & Associative Memories

Kohonen feature maps are used often as an associative knowledge representation scheme. They are nothing more than an interesting context sensitive representation method. However, much of the work presented here can also be implemented by using other knowledge representations. But the question is whether they will be as good.

The most impressive feature of the Kohonen maps is the automatic derivation of a topological map in such a manner that all related objects are stored in neighbouring areas. Moreover, the feature map derives conditional probabilities by incorporating much more context than simple Markovian features. When implementing other knowledge representation models, it will be a challenge to incorporate these two properties in the system, as they are mainly responsible for the required behaviour.

Problems with Kohonen Feature Maps

- Kohonen feature maps have a number of limitations:

- Kohonen feature maps are not really time sensitive.

- Kohonen feature maps need boundary effects in order to derive meaningful feature maps. Larger feature maps simply have relatively less boundaries and therefore derive qualitatively less well organised maps.

- It is hard to combine multiple feature maps in an elegant manner (although much research has been devoted to this subject).

- There is no (cheap) parallel hardware to implement Kohonen feature maps yet.

Extensions of the Kohonen model that address these problems should have high priority in order to use the maps in practical models.

Next, how does one interpret a topological map of words? This property of Kohonen feature maps is not used in the retrieval phase because one does not know how! This problem is closely related to some assumptions made about the underlying probability distribution. The Kohonen feature map requires a predefined network structure that should adapt to the underlying probability distribution (e.g., fixed dimension, fixed rectangular or hexagonal connection structure and fixed square, triangular, circular but continuous homogeneous topology). One of the main problems here, is that one does not know the form of the probability distribution of language. One thing is sure, it is definitely not two-dimensional, rectangular and homogeneous distributed (as is presumed by the form of the feature map used).

After the training phase, related objects must be in related neighbourhoods. However, what if a paper is on the border of multiple clusters. If this neurone is selected as the Best Matching Unit (BMU) on the Kohonen feature map, then the Euclidean distance does not represent a proper measure of correlation. One has to incorporate the cluster boundary knowledge in the classification decision. Such cluster boundaries must be derived by the model itself and not through an external supervisor.

Scalability of Kohonen Feature Maps

The scalability of the model remains a problem. Kohonen feature maps tend to entangle when they get larger. As a result, neighbouring neurones represent totally different objects. One method to avoid entangling is to train in a more context-sensitive way.

In [Kohonen, 1991], such a context sensitive training method is proposed: the Hypermap. This model is similar to the training algorithm presented here, however it trains global context properties first. As the model converges to a self-organising map of contexts, objects are trained to the map to obtain a fine-grained grid where necessary. By incorporating such a training method, the results as presented in this research are expected to improve also. A first simulation indicated that larger corpora (300 sentences) could be trained by using this method.

Other attempts to increase the feature map capacity can be found in [Kangas, 1992] and in the studies of hierarchical feature maps as mentioned in the previous two chapters. However, most results in this direction are still immature. Another, more promising solution can be found in [Fritzke, 1992a-b], who grows his feature maps from small to large in such a manner that the feature maps are always in order. As a result, the neighbourhood effects during the training will be less strong as they are in the original feature maps, but this can be solved by adapting the simulations. More on the importance of the capacity of the feature maps can be found in the next section.

Information Theory

Much has been written on the relation between neural networks and Shannon's information theory [Shannon, 1948, 1949]. A neural network can be regarded as a (multi-stage) encoder. Therefore, there is an obvious link between neural networks and information theory. In fact, notions from information theory can be used as a measure of performance of the neural network. More on these applications of information theory in neural network research can be found in [Bichsel et al., 1989], and [Kuhnel et al., 1990].

Information theory can also be used to explain other aspects of neural networks, such as the values, meaning, and relation with the natural sources of the sensory input. One of the first persons to do so was Ralph Linsker [Linsker, 1987, 1988, 1989a, 1989b, 1990]. In biological neural networks, sensory information is processed in a very efficient manner. Much of the natural data that is being processed is redundant. However, the human brain manages to remove most redundancies in an environment full of noise without the loss of information.

This notion is called the principle of maximum information and minimum entropy. Linsker showed that real Hebbian learning implements redundancy elimination without loss of information. He calls this the infomax principle, predicting the maximisation of the amount of information in the system.

Assuming L is an input signal vector and M is an output signal vector, then the neural network implements a mapping f: $M \rightarrow P(M|L)$ holds the conditional probability density function. The input probability density function is represented by $P_L(L)$ in case there is no feedback from $M \rightarrow L$, the output probability density function by $P_M(M)$. Then, in order to maximise the total information R, the following equation needs to be maximised:

$$R = \sum_L \sum_M P_L(L) \cdot \log(\frac{P(M|L)}{P_M(M)}) \qquad \text{(EQ 35)}$$

Next, presume that the points in *M* are distributed equally, and they are in topographic order. Then, if the neural network follows the infomax principle, the maps that are formed will have the following properties (see [Linsker, 1987] for details):

- The areas representing *M* will be equally large and uniformly distributed over the feature map.
- The map will conserve the topological distribution of the underlying probability density function.

Closely related to the original Hebb rule is the Kohonen feature map. On the one hand, Kohonen feature maps (empirically) tend to have the same properties as maps produced according to the infomax principle. On the other hand, some of the quantitative principles needed for the infomax principle such as lateral interconnections and topographic order are also present in the Kohonen feature map algorithm.

So, there is reason to assume that the weights which are derived in the Kohonen feature maps are in fact values representing the properties of the sensor values in such a case that the redundancy is eliminated without loss of information value. It is not stated that the weights in a Kohonen feature map are information theoretical values as proposed by Shannon and Linsker. The Kohonen model is much too limited with respect to the facts that:

- It does not maximise an expression such as the one given in the beginning of this section. Actually, it does not refer to any notion of information whatsoever.
- It is a local algorithm due to the local lateral interactions between the neurones. In the infomax principle, global interaction is presumed.
- It does not consider aspects such as noise.

There is a clear relation between the two models, indicating that the Kohonen feature map does converge to something more valuable than plain frequencies. Due to the complicated neighbourhood interactions a set of weights is derived of which each represents an element of the training set with respect to the frequency of occurrence as well as to the context in which it occurred, just as it is the case in information theory.

So, Kohonen feature maps, as they are used in the simulations, derive conditional probabilities of the corpus fragments. Although one cannot give a quantitative measure of the amount of conditionality, it should be possible to indicate at least a lower and an upper bound of the context that is incorporated.

The Upper Bound

The conditionality is derived during the training process by the neighbourhood effects. If the feature map is in perfect order (that is, all objects are organised in such a manner that related objects are in neighbouring areas), the amount of conditionality will approximate the information theoretic values as indicated by Shannon because all elements are stored with respect to all the proper contextual influences that exist in the system.

The Lower Bound

However, for a perfect organisation, Kohonen feature maps require that the form of the Kohonen feature map follows the form of the underlying probability distribution. Here, a two-dimensional, rectangular feature map is used, presuming that natural language is two-dimensional and rectangular. As all can see, this is a very rough simplification of the real world situation. Due to this limitation, and due to the fact that the feature map is not in perfect order during the training process, the values that are derived by the neurones do not incorporate all possible contextual influences and are therefore less good than information theoretic values. If one presumes that the feature map is never in perfect order, the values that are derived incorporate only the context as given by the shifting window: simple Markovian conditionality, which can be considered as a lower bound of the value of the probabilities derived by the feature map.

So, the probabilities that are derived by the feature maps are better than Markovian conditional probabilities, but they are worse than Shannon information theoretic values. As a result, the more context sensitive and the less entangled the feature maps are during the training process, the better the probabilities are that they derive.

Clustering with Kohonen Feature Maps

Here, it is stated why Kohonen feature maps and other clustering algorithms might not be suited for clustering. Once the vector has been determined, it can be fed forward to the feature map to derive the Best matching Unit (BMU). This BMU represents the document most correlated with the vector. The neurones within a certain Euclidean distance *d* hold related documents. However, one needs the BMU as well as the cluster boundaries to make a responsible decision on the measure of correlation between documents.

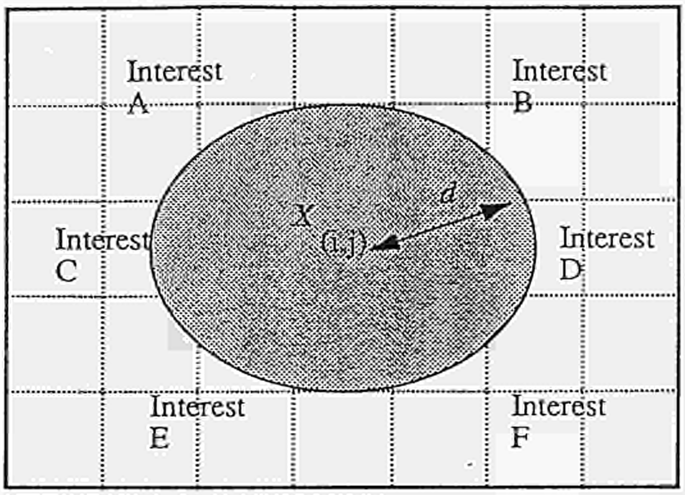


FIGURE 6.1: AN IDEAL FEATURE MAP.

If the vector *x* correlates best with the BMU with weight *w* at neurone *(i,j)*, then all neurones within distance *d* are supposed to be related (dark circle). But, what if the neurone at the feature map looks like the one derived in "An Ideal Feature Map". If the BMU seems to be the neurone at position *(i,j)*, it is positioned exactly at the border of multiple interests. The reason why these interests are neighbouring is not because they are related, but because they are forced to interconnect due to the dimension reduction properties of the feature maps. If one uses the Euclidean distance as a selection criterion, the documents selected are not the proper ones. Some interests which are neighbouring have nothing to do with each other.

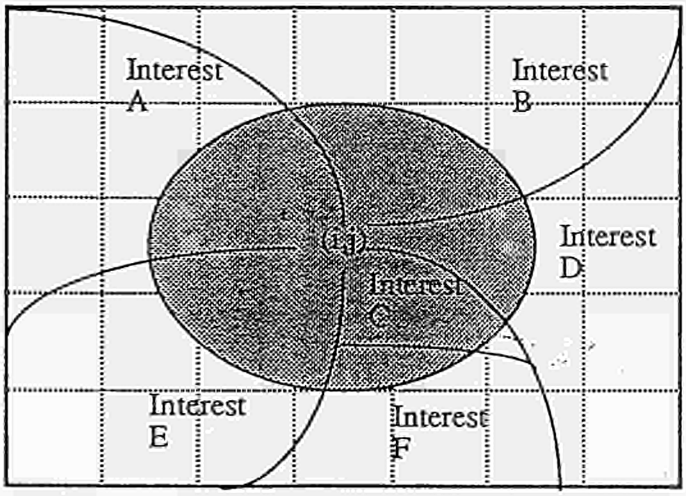


FIGURE 6.2: A FEATURE MAP AS THEY OCCUR IN THE REAL WORLD.

So, the only way to make a reasonable decision is by incorporating the BMU as well as the cluster boundaries. This is a big disadvantage, because then one has to determine the cluster boundaries manually. These decisions take much work, are very personal and therefore subjective and sensitive to errors. Therefore, one should be very careful with these kind of clustering algorithms.

Kohonen Feature Maps, Back-propagation and Other Neural Paradigms

Is the Kohonen feature map the best neural model for the simulations carried out? There are many other neural models. The early neural information retrieval used localist knowledge representations (one neurone for one concept). Recent efforts showed the application of feed-forward and recurrent back-propagating networks. Kohonen feature maps are just recently invoked in IR applications. Hopfield networks and other associative memories have also been used, but only rarely.

In general, clustering and generalisation problems can be solved by self-organising networks, such as the Kohonen feature map, the Simple Recurrent Network (SRN), and ART. Mapping problems or function approximations can best be done by a feed-forward back-propagating neural network. Temporal processing can best be done by a SRN or any other recurrent model. Associative memory problems might be solved be either neural network: BP, Kohonen, ART, etc. Of course, these applications can also be solved with other network types, but the networks mentioned are the most natural choices.

Information Retrieval is a clustering problem. Based on a selection of specific features (e.g., n-grams or keywords), a representation of an object is derived by feature extraction. The objects are categorised in clusters by the retrieval function. The main issue is the determination of such features, so the difference between clusters is as big as possible (or, as little overlap as possible, since overlap causes the classification error). Because the Kohonen feature maps are the computationally most effective self-organising networks, they are in fact the best neural network for such problems. Once more, one does not want a supervised model, because it is not known what to learn in the first place.

However, it is also possible to use a SRN to teach a representation to the neural filter. A disadvantage of the application of a SRN in the neural filter is the fact that the model implements an high order Markov chain by using recurrent fibres. This is just much too sophisticated. If one uses a regular BP network with a window, the network does not form a representation as good as the Kohonen feature map. The representation is much more discontinuous.

Moreover, it is difficult to structure the input set. In the case of the neural interest map, either the SRN or the Kohonen feature map does well. Both have shown to be pretty good in such clustering problems but both run out of addressing space. An advantage of the Kohonen feature map might be a faster and more stable convergence although they have not been used for large (10.000 neurones) feature maps. Recent simulations of SRN's in IR showed very long training times and unstable behaviour for large data sets. Advanced training techniques may suppress these effects for a while, but they remain deadly in the long run [Elman, 1991a-b].

If one wants to learn a specific mapping or function approximation, then back-propagation seems to be the best choice. However, one has to realise that most IR problems are clustering problems and not mapping problems, making BP a second choice.

6.4 Information Retrieval

Two different types of information retrieval can be observed. On the one hand there are relatively static databases which are investigated with a dynamic query (free text search, also known as document retrieval systems). Next there are the more dynamic databases which need to be filtered with respect to a relatively static query (the filtering problem also known as current awareness systems [DARPA, 1991]). In the first case, data can be pre-processed due to their static character. In the second case, the amounts of data are so large that there is no time whatsoever for a pre-processing phrase. A direct context sensitive hit-and-go must be made.

The localist models adapt well to the models currently in use in information retrieval. Index terms can be replaced by processing units, hyperlinks by connections between units, and network training resembles the index normalisation process. However, these models do not adapt well to the general notion of neural networks for the same reasons as is the case with local connectionist NLP models.

In addition, it is difficult to imagine what to teach to a neural information retrieval system if it is used as a supervised training algorithm. The address space will almost always be too limited due to the large amounts of data to be processed. A combination of structured (query, retrieved document numbers) pairs does not seem plausible either, considering the restricted amount of memory of (current) neural network technology. Nevertheless, most of the neural IR models found in the literature are based on these principles.

Also problematic are the so-called clustering networks. Due to the large amounts of data in free text databases, clustering is too expensive and is therefore considered irrelevant in changing information retrieval environments [Willett, 1984, 1988].

In general, more interesting are the more unsupervised, associative memory type of models, which can be used to implement a specific pattern matching task. This type of neural networks can be particularly useful in a filtering application. Here, the memory demands of the neural network only need to fulfil the query (or interest) size, and not the size of the entire data base. It is here where neural networks are expected to be useful and relevant for information retrieval.

6.5 Neural Networks for Information Retrieval & Information Retrieval for Neural Networks

Information retrieval, being a clear pattern recognition problem, has mainly benefited from statistical pattern recognition technologies. The enormous amount of data to be processed actually did not allow any other methods within practical limitations. As time passed, many researchers tried to increase the level of analysis without blowing up the computational needs. Due to this constraint, the information retrieval tool box could never use any linguistic theory. Therefore, the algorithms used are restricted to local surface analyses.

Recent research in connectionist natural language processing showed interesting self-organising models that can learn approximations of finite state grammars and simple semantical relations from unformatted data. Moreover, these neural devices showed remarkable competence in clustering and classification tasks of incomplete data sets. All these properties are well known functional demands for information retrieval systems. Maybe that is the reason why the number of papers appearing in literature is increasing so rapidly.

Besides the positive contribution from neural networks to information retrieval, there is also one the other way around. Information retrieval has a long and well understood history in statistical pattern recognition. Many problems have indeed been solved by using statistical methods. Comparisons of such results with new results in neural information retrieval open doors to a better insight in the exact relation between neural networks and other classical pattern recognition solutions. Because, if neural networks are such good pattern classifiers, where does one position it with respect to the known pattern recognition theories?

Even more interesting is the contribution of information retrieval to NLP. Because information retrieval problems are often much simpler, they clarify neural bottle-necks much easier than NLP problems, thus contributing to the development of better neural models for NLP.

6.6 Neural Networks as Hashing Functions

If one studies the behaviour of the some neural nets in information retrieval, the question arises as to what its exact relation is with another well-understood addressing technology: namely hashing [Boyer et al., 1977], [Bozinovic et al., 1982], [Harrison, 1971], [Knuth et al., 1977], [Larson, 1988], [McIllroy, 1982].

On the one hand, the relation is very clear: neural networks are large (calculating) associative memories, able to store elements efficiently. On the other hand, the reason why certain elements are stored and others are eliminated is not yet clear.

6.7 Higher Order Linguistics & Knowledge Representation

A major problem in information retrieval has been that higher level analyses were not available at reasonable prices. Recent research in connectionist neural networks showed how to teach approximations of finite state grammars to recurrent neural networks. By training word sequences, neural networks were able to learn regular grammars. Although these grammatical structures are the simplest possible, they can definitely increase recognition performance. However, the recurrent models learn very slowly and are quite unstable. It seems that most information retrieval systems are aided sufficiently with a restricted Markov model (such as trigrams over words). The fact that neural networks can learn these relations in linear time, opens up new possibilities for neural networks in information retrieval.

So much for the treatment of structural analysis in IR. Another important problem is to incorporate meaning in IR. Yet, there is no real meaning involved in the algorithms proposed. There are only the contextual relations incorporated in the n-gram representation. By generalising over these contextual structures, simple semantic relations can be derived. However, real meaning and the interpretation of conceptual structures is more complicated, and should not to be taken lightly or solved solely by means of n-grams.

Other research focuses on knowledge representation structures for information retrieval. The early connectionist models were mainly used for such applications. Only recently have neural networks been used for clustering tasks. A possible use of such clustering neural networks for knowledge representation is in the application of hierarchical feature maps, where relations between objects and classes of objects are captured in the hierarchical structures. Another solution can be found in [Allen, 1991], who uses a simple recurrent network (SRN) to teach semantical issues to a back-propagating network.

6.8 Applications

In this section, the constraints for the application of a neural network in information retrieval will be given per application.

- Library management

  In general, library management problems have a very structured character, because nobody wants to loose a book to the free interpretation of the neural network. As mentioned before, neural networks are not very suited for such a task. They are much too sloppy. On the other hand, there are some characteristics in the task of a librarian that require advanced predictive capabilities. It is here that neural networks can assist in predicting loan requests per title as a non-linear time series problem.

  The data required for this task can be obtained automatically from user behaviour and the model can adapt over time, without the users noticing anything.

- Information clustering

  Clustering of information can be done on the basis of bibliographic information (titles, author, etc.), keywords, titles, abstracts or even the full-text. In all cases a number of problems cannot be solved properly yet. First, neural networks have not been shown to be able to handle the large amounts of data that are involved in typical information retrieval applications. Second, neural networks cannot derive and store sharp hierarchical structures that can be derived by statistical cluster methods such as branch-and-bound and k-nearest neighbour methods. Third, clustering is an expensive process, mostly of quadratic order of the number of elements in the data set.

  Despite those problems, neural networks seem to implement an interesting clustering device for thesaurus generation, hyperlink derivation, associated word derivation etc. However, it is the opinion of the author that none of the currently presented neural clustering models does a good job on large data sets. Therefore, the successful application of neural networks in clustering problems is doubted.

  If on the other hand, the data sets would be considerably smaller, and if the input would be natural (noisy) input, then a neural network could be used as a feature extractor to minimise or filter the data input. In such a case the obtained data sets could be used by a classifier or other mechanism. Then, a neural network as information front-end might be useful

- Interface design

  There are two directions in interface design. The first is the derivation of hypertext structures. The second is an extension of the search engine with an adaptive module.

  In the first case, all models presented use localist models that are not very interesting from a neural point of view. Naturally one can add knowledge to a system by designing a hypertext or conceptual structure on top of a indexed data base. But this has nothing to do with neural networks. Traditional hypertext structures and thesauri are good enough for this job.

  In the second case, localist as well as distributed models can be seen in literature. On the one hand, localist models are not very interesting for reasons mentioned before. On the other hand, by training a neural network with previous input-output results, a model that adapts itself to an application or data base is obtained. Nevertheless, the address space of a neural network is never large enough to store all of the possible input-output pairs (or store even a small subset of the data). Therefore the practical application of these models can be doubted.

  An interesting variant of the second case is the application of data fusion where several relevance ranking values are combined into one overall value. This very non-linear and domain dependent mapping can be trained easily to for instance a back propagation neural network. Results in literature have shown retrieval improvements of 40% with respect to the best relevance ranking technique (where one does not know what the best technique is).

- User Modelling

  One of the applications in which neural networks are expected to be successful is the field of user modelling in current awareness and SDI. First, only the user interest model is stored in the neural net. As this is a limited data set, one does not have to solve the memory issues mentioned before. Next, neural networks are good at implementing noisy, heterogeneous abstractions of text. That is exactly what one needs in user modelling.

  In addition, the user model can adapt automatically to a change of interest, this could even be done by providing a relevance value to a "typical text" representing a certain interest model or concept.

One should be aware of the fact that the user models will only implement the above described properties if they are distributed, self-organising, and massive parallel!

- Incomplete Searching

  The application of neural networks for the matching of incomplete data sets is so obvious that all commercial applications available for this task use neural networks. It is here were traditional statistical pattern recognition techniques can do a good job, but neural networks will always be more robust.

- Searching in Multi-Media

  As multi-media is almost always noisy, neural networks are one of the few options in this field of application. The problem lies more in the abstraction levels allowed in the multi-media query than in the "neural matching" algorithm.

  Main problem here is the limited memory capacity that neural networks have. Therefore, one is probably able to store the query in the neural network, but not to store the data base in the neural network. That means, that the entire data base has to be passed along the query as a search is performed. This can be a serious limitation.

- Data Mining

  Case based reasoning and data mining are two applications for which the typical neural information retrieval algorithm are of great value. Combinations of different information sources, generalisation, association and noisy matches are absolutely needed in such systems. By extending the application as described above, the authors expect many possibilities for "real" neural networks in this area.

  However, the main problem will be the lack of address space that a neural network has. The used models should either use a very smart encoding scheme or limit the usage of neural networks to those parts of the system that do not need such large amounts of memory (such as the query, an interest model, or a concept).

6.9 Proposed Research in the Prototyping Phase

Based on the literature study and on previous work by the author a number of directions were identified as worthy of further study. It was felt that additional improvements could be made in the way that the properties of certain types of ANN's were being used. Secondly, in many cases the studies reported in this chapter were preliminary pilot studies. Often these studies identify interesting ideas, but it remains unclear how these ideas will stand up against real-world application demands. In all cases the results of our prototypes will be compared to the best available traditional techniques. The prototyping phase is described in the next part of this report. The following directions will be investigated:

- Fuzzy search with an (extended) Kohonen network on typical OCR errors on several hundreds of OCR-ed documents;

- Adaptive user modelling for information filtering & current awareness with a Kohonen feature map on over 500 Mbyte of data.

- Clustering & interface design with variants of the Kohonen network on large databases of bibliographical records.

The first two applications were expected to be very successful as they involve the storage of a small amount of vaguely defined information in the neural network. There were doubts about the feasibility of the third application, *viz.* clustering, but the application was seen as one of the most intuitively appealing uses of ANN's in IR. Therefore it was considered a worthy candidate for a thorough empirical test.

Part 3

Prototyping and Experimentation

7 Motivation

"I may not have the answer,
but I think I've got a plan"

-- Jackson Brown

In this chapter, the projects pursued in the prototyping phase are introduced and a motivation is given for the choices made.

The survey of the literature and the insights about the functionality of ANN's (Artificial Neural Networks) as presented in part 1 and 2 of this study, combine to give an interesting perspective on the possibilities of the use of ANN's in IR (Information Retrieval). The following insights formed the motivation for the areas pursued in the prototyping phase.

Some proposed applications seem to be unrealistic, particularly due to the huge difference in scale between the real world retrieval task and the toy domain for which performance was investigated. In the State-of-the-Art report a number of promising or controversial areas were identified as well. These were considered worthy of further research in the prototyping phase.

Some applications seem to have the potential to become very successful, especially those where the neural network is used to extract small vaguely defined information patterns from large amounts of noisy data. Sometimes there are no well established alternative methods, because the problem is as new as its solutions. This situation, however, is not present in many typical library IR problems. The storage of user profiles for the filtering, routing or selective dissemination of information, and the storage of typical error patterns for fuzzy search or for the correction of corrupted OCR documents can both be classified in this category. For user profiles ANN's promise advantages like content addressable memory and associative definition of vague concepts. For the storage of error patterns their main advantage is that an unknown function, *viz.* the nature of a stochastic error source, can be learned from incrementally presented examples.

A number of the applications reported in the literature fall into a grey area. The use of neural networks for tasks like thesaurus construction, document clustering and interface building seems to be very interesting at first glance. For these kind of problems, which are essentially clustering problems, well established statistical methods often exist, but neural networks introduce a powerful new perspective. In many cases in the literature superior performance, or entirely new functionality is claimed, based on ANN properties such as automatic generalisation, graceful degradation and parallel computation. In fact this means that ANN's are claimed to outperform classical methods on counts of effectiveness and efficiency. However, such has hitherto remained largely without empirical support.

In this part of the report, the implemented prototypes will be discussed in detail. Due to the nature of the various tasks, the studies presented below differ in their character. The prototypes in the areas identified as likely to be successful implement a fully working system. Many of the practical issues encountered in the process of building these prototypes are discussed. In the chapter about the clustering prototype, the claims in the literature are further examined and an answer is sought as to whether these claims are justified. The following prototypes shall be described in the next three chapters:

- The Fuzzy Search Prototype[5] is described in chapter 8. This prototype examines the effects of fuzzy search on a large text database which was created by Optical Character Recognition (OCR). Traditional fuzzy search algorithms use wild card operators to ensure a reasonable degree of recall. However, wild cards are too general; they do not take into account that the errors introduced by e.g. OCR are not random. In chapter 8 it is shown that training a search algorithm on a large amount of typical OCR errors improves performance considerably. ANN's are compared to statistical confusion matrix methods on this task. The experiments show that the neural network has a positive effect on performance, although the size of this effect remains quite limited.
- In chapter 9 the Clustering Prototype[6] is discussed. This prototype explores the feasibility of the concept of "semantic road maps", i.e. browsing interfaces for information visualisation and retrieval which are derived by document clustering. The resulting interfaces are supposed to be like maps in the sense that distance in the map reflects semantic similarity. It is shown that a number of the problems which have been identified in the state-of-the-art report can be solved by employing an extension to the Kohonen map, the Growing Cell Structures Network of [Fritzke 1992]. An experimental comparison shows that ANN's perform at least as well as traditional clustering methods on this task.

[5] The implementation for the Fuzzy Search Prototype and the experiments were performed by Micha Leuw.

[6] The implementation for the Cluster Prototype and the experiments were performed by Jakub Zavrel.

- The Filter Prototype[7], described in chapter, applies a Kohonen network augmented with sophisticated speed optimalisations to filter huge amounts of text fast. The neural filter, tested under a wide variety of parameter settings, seems to perform very well.

Each of these chapters will describe the neural algorithms used, the results in an ideal environment and the results in comparison with a traditional method in a real-world environment. For each of the prototypes, directly related conclusions will be given in each chapter. More general conclusions with respect to neural networks in IR are discussed in *Part 4: General Discussion and Conclusions.*

[7] The implementation for the Cluster Prototype and the experiments were performed by Marco-René Spruit.

8 Fuzzy Search Prototype

-- Micha Leuw

This chapter describes the efforts made in improving a traditional fuzzy search method for OCR-errors by using a Neural Network and a probabilistic confusion matrix. The results of these three methods will be compared qualitatively and quantitatively.

8.1 Introduction and Problem definition

Optical Character Recognition (OCR) seems to be a relatively simple task. Different characters have different forms. And even the difference between tokens of similar characters (e.g. e and c) is, although very small, recognisable with a great degree of accuracy; for the human eye at least. Already the first commercial recognition software for hand-written text are hitting the streets. Therefore, many laymen assume that recognition of printed text must be perfect.

However, this is not the case. Although the performance of OCR-software is improving rapidly, errors are still made [Rice et al., 1992-1994]. Reasons for these errors are diverse:

- *Degraded originals*: The hard copy, the physical text page, might be degraded: for example, it is copied to often, printed badly, or printed on paper of low-quality.
- *Differences in character size and font.* That is, some fonts make recognition very hard.
- *Zoning errors*: OCR software needs to know which parts of the scanned image should be interpret as text and which part contains graphical information, this process is called zoning. As well automatic as interactive zoning is error sensitive.

8.2 Background

Presently more and more text is scanned and used in full-text databases. It is estimated that correcting OCR-text costs from $2 up to $5 per page. Since we are dealing with such large amounts of text (often > 1,000,000 pages), correction would be way too costly.

Therefore current full-text database packages like Excalibur and ZyIMAGE enable an alternative strategy. This strategy is based on the principle that no correction is needed if the user can be presented with a readable version of the text. These packages store the image files, the graphical representation of the hard copy, in combination with the OCR generated text-file. Links are kept for each physical page between the text and image-files.

If the quality of a certain text-page is to poor to read properly, these links make it possible to pop-up the image file, so the "real" text can be studied, including all extra graphical information that is lost during the OCR-process such as graphs, tables, logo's, formulas and signatures.

Need for fuzzy search

If the OCR-ed text is used this manner, the need arises for good retrieval tools. Some text files might still contain OCR-errors and the user is surely interested in these files too. A normal keyword-based search engine would miss these files since they do not contain the exact keywords. This introduces the need for a *fuzzy search* mechanism: a mechanism that expands a query by adding possible alternatives of keywords to the query.

Neighbourhood Effects

Since all fuzzy search involves probable character changes from the source word to a possible occurring miss-spelled OCR-variant, there is a chance that not only miss-scanned occurrences of a source word will be found, but also different, unrelated, words. As a result, the search precision will be lower. This problem occurs most often for words that have many orthographic neighbours, i.e. a lot of the generated permutations are other existing words.

For instance if a word like 'procent', Dutch for 'percent', is searched and one would allow the search mechanism to change two characters, words like: 'recent', 'percent', 'project', 'present', 'process' and 'precent' will be found. By contrast, changing two letters in the word 'Philips' is relatively safe.

Therefore 'procent' is said to have more orthographic neighbours than 'Philips'. In general, small words have more neighbours. I.e., changing two letters in a two-letter word would probably produce another two-letter word. While changing two-letters in a 12-letter word would probably produce a spelling variant of the original word and would only in very few cases produce another word. For this reason, fuzzy search is pretty safe on large words and virtually impossible on small words.

Goal

Therefore, the efforts made are aimed at improving the precision of fuzzy search for short words. This will be done using two techniques:

- *The Confusion Matrix* method, a simple statistical technique, based on occurrence counts of OCR errors actually occurring in a realistic corpus.

- *A Neural Network*, trained on examples of OCR-errors.

Both techniques have in common that they will restrict the interpretation of the joker (?) used in the traditional method, called *Wild Card* search, by using "knowledge" of typical OCR-errors.

8.3 Wild Card Search

The simplest fuzzy search method is Wild Card search. This method is implemented in the retrieval program ZyIMAGE which has been used for all the testing and comparison. A wild card search method uses three basic character operations: insertion, replacement and deletion. Any character of the source word can be deleted or can be changed into any other character. In addition, it is possible to insert a new character between each two characters and at the beginning and at the end of a word.

The maximum number of mutations allowed to be made on a source word is known as the fuzzy degree. For example a wild card search on 'cat', with fuzzy degree one, will find, among other things, instances of the following search templates: '?at' , 'c?t' , 'ca?', 'at' and 'c?at', where '?' stands for any letter.

Using an unrestricted (any letter to any letter) replacement regime has certain advantages

- Semantically related words are found. For instance a fuzzy search on the word 'application' will find 'applications'

- Misspelled instances ('application') and typing errors will be found.

- No knowledge about typical OCR- or typing-errors has to be stored.

However, as the figure below shows, a wild card search does a very bad job on short words. There seems to be a critical fuzzy-degree for words of a certain length. If the fuzzy degree is set higher than this critical level, an explosion in the number of hits can be observed. The program returns too many files to be in any sense useful. For longer words this problem doesn't exist. E.g., the fuzzy degree can be set to three in almost all cases for words longer than 7 without causing a collapse (see Figure 8.1 below).

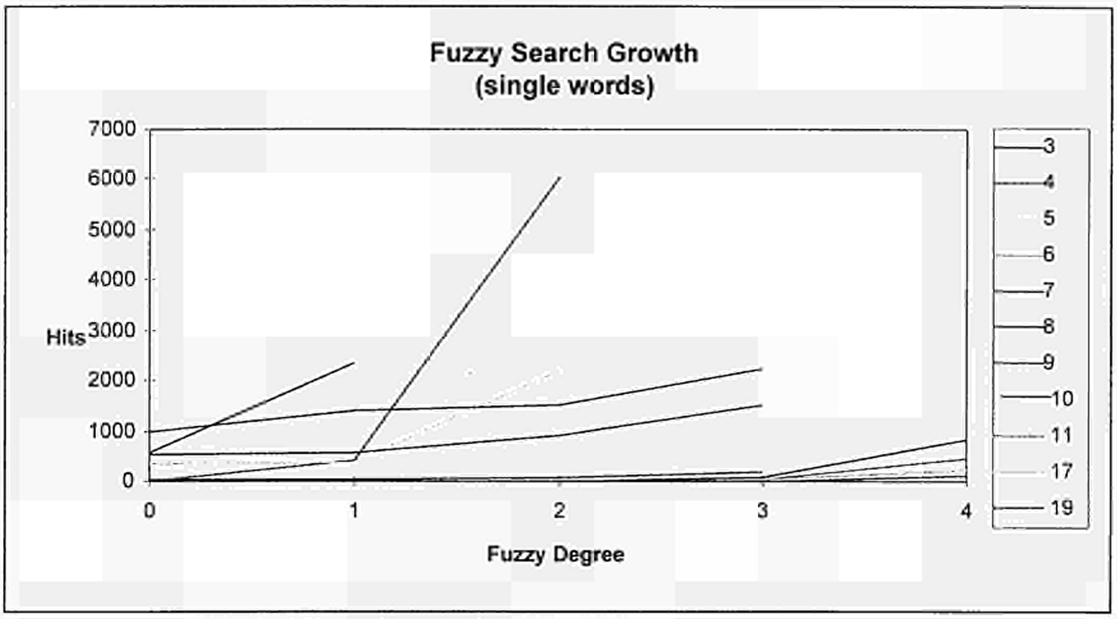


FIGURE 8.1: THE FUZZY SEARCH GROWTH IN KEYWORD PERMUTATIONS WITH RESPECT TO THE FUZZY DEGREE

A wild card search has to use an index to restrict the search-space. In principle the generation process results in a combinatorial explosion of the possible number of mutated words. Because a wild card search will also generate many words that are not in the text, it would be computationally very inefficient. Using a full-text index makes it possible to guarantee that no paths are searched that would lead to any non-existing word. However, one should maintain a full-text index of the entire database to use this optimization method.

One of the major setbacks of wild card search is the fact that it doesn't provide any likelihood ranking of the words that are found. That is, no probabilistic knowledge whatsoever is used to generate and order spelling alternatives. Therefore, all generated alternatives have to be checked, whereas in an ordered generation, only the most relevant alternatives have to be looked up. Because of this, it seems fair to assume that a more intelligent algorithm might do better in small words.

The Data set

For the two fuzzy search methods a set of typical OCR-errors was needed. By comparing the original versions of Part Two of this report: *The State of the Art Report* and [Scholtes, 1993] to their OCR-generated ASCII-files using a string-matching algorithm 1743 errors were obtained (from a total of 500 pages of text). The string matching algorithm used works as follows.

- Take two strings as input: the original (source) string and its OCR'ed counterpart (target)
- Delete the matching beginning and ending of the strings.
- Until both strings are empty do:
    - Find and delete the longest common substrings, and recursively proceed with the left and right parts found this way.
    - If no common substring can be found store the strings as an "error".

It's very important to notice that the OCR-errors, hereafter called transitions, found by using this string comparison algorithm are mappings of character combinations to characters combinations rather than to individual characters. In other words, these are mainly n to m mappings instead of 1 to n or n to 1 mappings.

In many cases transitions were of the form nn→m or h→li.[8] The relation underlying the OCR-errors can be seen as a many to many mapping of sequences of zero or more characters. This point will be very important in the design of the representation scheme later on. We shall call such sequences *symbols*.

After deriving the confusion matrix, which is simply a list of all observed transitions, some analyses was done. This was done for two reasons.

- First, these analysis will set the limits to the alternative methods.
- Secondly, these analyses will play an important role in the process of reducing the data set in a later phase.

The next figure provides the reader with an insight in the distribution of the errors.

[8] The symbol → is read as: "goes to".

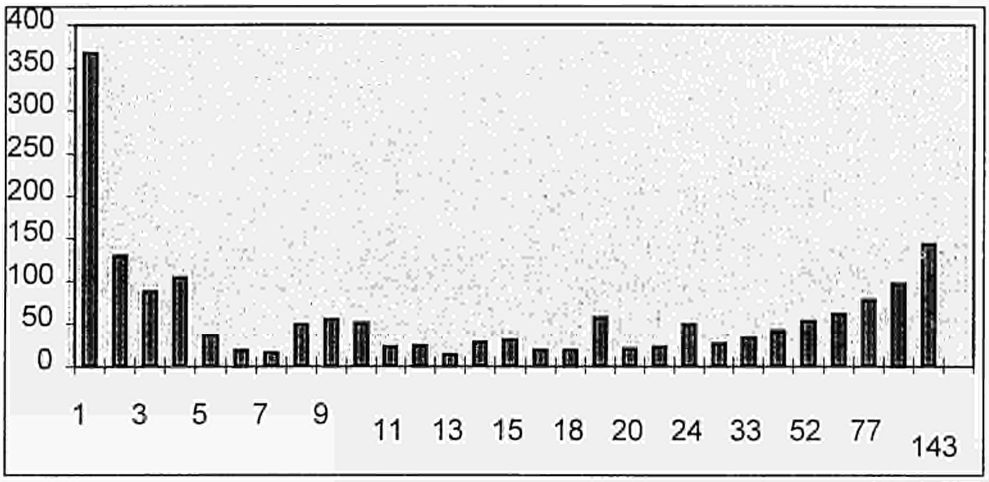


FIGURE 8.2: THE DISTRIBUTION OF THE ERRORS WITH RESPECT TO THE TOTAL NUMBER OF OCCURRENCES OF AN ERROR

The x-axis gives the frequency group of transitions types, i.e. the count of a certain error. On the y-axis the total number of transitions within that group is plotted. So, the total number of errors that occurred only once is 315. The single error with the highest frequency, (rn→m) was made 143 times.

This distribution is very important since it shows the limits of the use of a restricted replacement scheme or any statistical method beforehand. Transitions that lie in the low frequency area are unsystematically error's so they it will be very hard to use them. High frequent errors show principal shortcomings of the recognition software for certain symbols. Therefor the more of these high-frequent errors one has, the better.

Within this data-set approximately 25% of the errors found have a frequency lower then three. Manual inspection of these errors shows that most of these errors were deleted text parts and other semi random errors, i.e. ([Deffner → VOID Where VOID denotes the empty string).

Further analysis was done on the distribution of source and target symbols. This provides us an idea which symbols are hard to recognise, which symbols are often confused with other symbols and what the distribution of these is. This data will be used in preparing the data for the neural network.

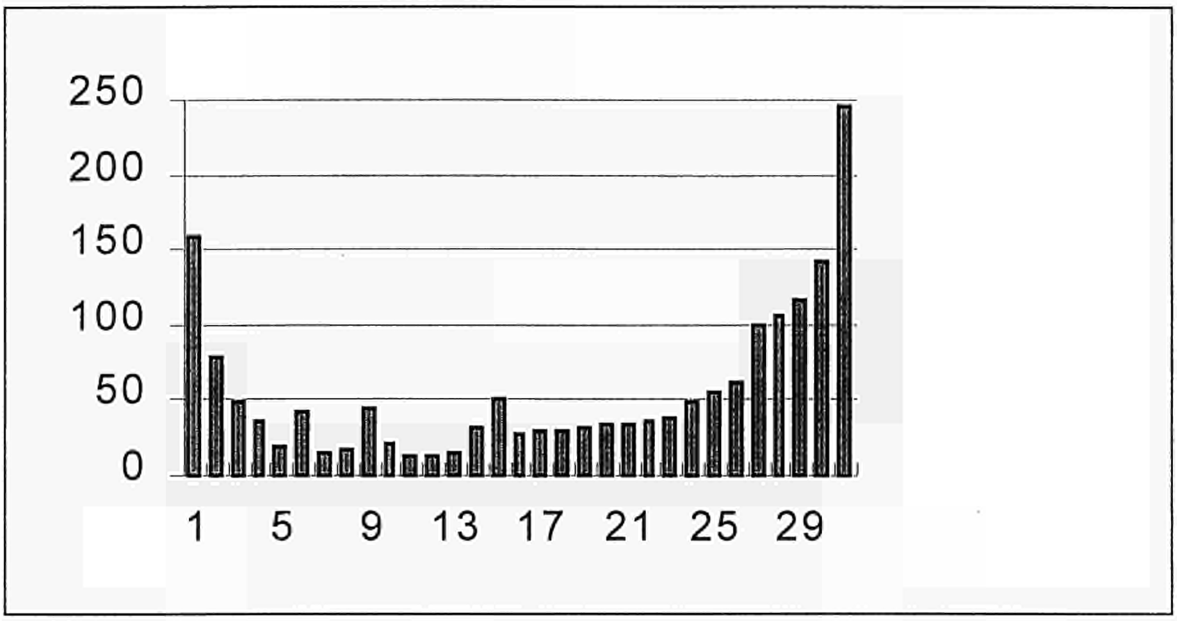


FIGURE 8.3: DISTRIBUTION OF THE SOURCE SYMBOLS.

From Figure 8.3, it can be concluded that there exists a large group of source symbols that is often misrecognised. So there is a group of symbols that is hard to recognise for the OCR device. On the other hand there is a large group of symbols that is miss-recognised just once. This will turn out to be a very nice property later on.

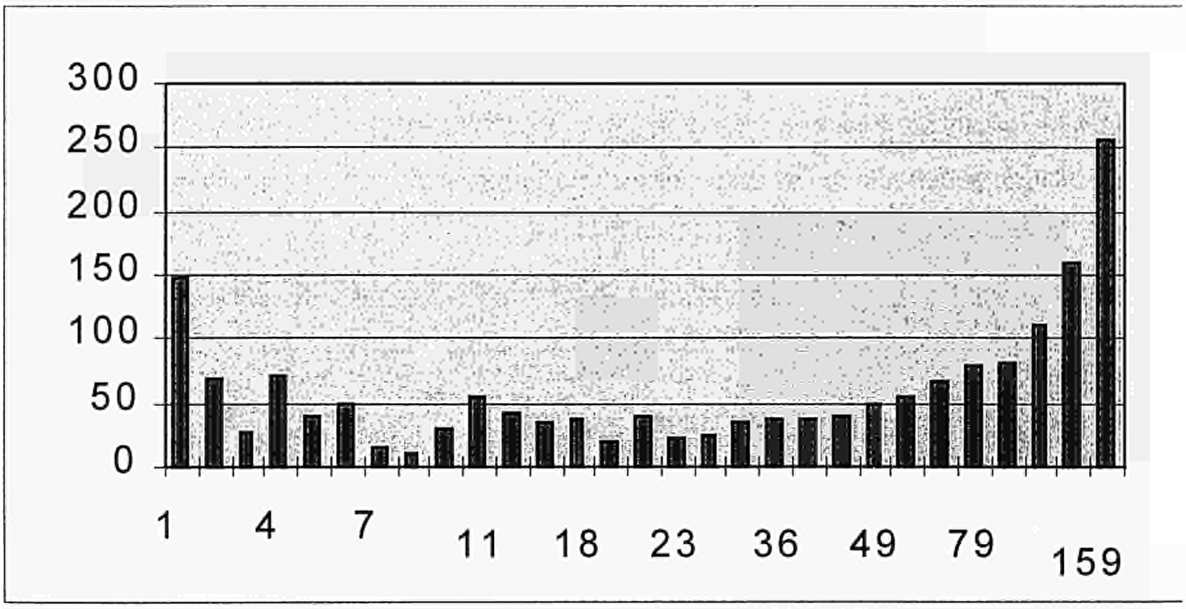


FIGURE 8.4: DISTRIBUTION OF THE OUTPUT SYMBOLS.

As can be seen in Figure 8.4, the same holds for the target symbols.

8.4 The Confusion Matrix

The confusion matrix method uses the set of transitions, the so-called confusion matrix, in a direct manner. The general idea behind the use of a confusion matrix is that only those transitions that have been observed should be in the confusion matrix. Only these transitions, and no others should be used to modify the source word for fuzzy retrieval. In other words: the confusion matrix is used to change the interpretation of the 'joker' (?) used in the wild card search in a twofold manner:

- First of all the joker no longer matches just single letters but symbols.
- Secondly, the possible symbols the joker can be changed into is heavily restricted.

To illustrate these points. If a word like 'learning' is searched for and the confusion matrix contains the transitions rn→m and rn→iii and no other transition for the source symbol 'rn' then only the words 'leaming' and leaiiiing' will be added to the query. Compared to a wild card search this is much more efficient. To find 'leaiiiing' fuzzy degree should be set to three, i.e. two replacements and one insertion are needed. So the wild card search engine would also try to search for words like 'leaking' or 'fetqning', among many others, resulting in longer search times and lower precision.

Another advantage of the use of symbols over characters is that a larger part of the source word can be changed with fewer direct character mutations. A word like 'teaming', which is one of the OCR-errors found during testing, is generated by the confusion matrix method doing only two mutations while Wild Card search would need a fuzzy degree of three.

The big disadvantage of this method is that it doesn't find semantically related words, and spelling errors not caused by OCR, as Wild Card search does. However the error model could easily be extended to cover these types of permutations as well.

Probabilistic Confusion Matrix

A straightforward implementation of the Confusion Matrix uses only the absolute frequencies of the transitions. A transition is assumed to be more likely than another one if it occurs more often in the training-set. In a way, this is a very naive approach since it does not take the a-priori probability of a certain source symbol into account. For instance the letter 'c' was miss-recognised as an 'e' in six cases while as 'q' was seen as a '6' in only three cases.

After normalisation with the a-priori source symbol frequencies. It turned out that c→e was less likely than q→6. The normalisation step makes the confusion matrix a probabilistic model for OCR-errors.

The Generation Process

The search process used in the implementation of the confusion matrix method was different from that of the wild card method. In the wild card search, generation of possible alternatives and the actual search for these alternatives are intertwined by the use of the B-tree represented full-text index. Our confusion matrix implementation uses two sequential steps: first generate alternatives, then look if they exist.

This reintroduces the combinatorial explosion of alternatives problem. To solve this problem and at the same time get relevance ranking, the probability of the generated alternatives was used so that the generation would produce as output only the *n* most likely target-words. The best way to do this would be to find an algorithm that generates target words in order of likelihood. This turned out to be very hard. The varying symbol length and the nature of the mapping (especially the fact that it's a one-to many mapping for a given source symbol) were the complicating factors.

The algorithm that is used can be described as a breadth-first generation process with pruning on probabilities of partial results. First all single mutations of the target word are generated to fill an array which is used to hold the alternatives and make pruning more efficient. Then the search tree is traversed. If the number of *n* desired alternatives is reached and the probability of a current partial result is lower than the minimum probability of the most unlikely generated target word, then this branch of the generation process can be pruned. This algorithm turned out to be efficient enough to get good search speed.

8.5 The Neural Network

What could one expect from the use of the neural network for fuzzy search? Neural networks are, in general, model-free probability estimators. One might think that in the current case the probabilities do not have to be estimated anymore since the probabilistic Confusion Matrix is the best available probabilistic model anyway.

In a sense this is true. Artificial neural networks, from the author's point of view, suffer from a dilemma. They are hard to use in real large scale problems. Most neural network have a scalability problem which might be due to the fact that the error-surface becomes more complex for greater data sets or to computational reasons.

The big advantage of neural networks is that they build a probability model in a completely automatic fashion. Because of this, neural networks are often said to provide the second best solution to any problem. Which might be good enough if the data set is too large to analyse in any other manner or if the data set is not well understood. However most small scale problems can easily be (automatically) modelled by some traditional statistical method and have data-sets that can be well understood.

So, neural networks might have some application prospects on data-sets that are big enough to make them hard to solve by hand and that are small enough to fall within the scalability boundaries of neural networks. Whether this "area" contains some interesting problems remains an open question. Fuzzy search for OCR-errors clearly is a problem that is small enough to be implemented with alternative statistical methods, e.g. the confusion matrix.

Using a neural network, however, has two advantages:

- First of all neural networks are good in forgetting what is unimportant. Transitions that are seen very seldomly in the training phase will be forgotten.
- Secondly, they provide a natural way to incorporate context sensibilities in the search process.

Desired properties of the Neural Network

What properties should our neural network have? First of all, it should enable us to retrieve OCR-errors and something like an associated likelihood ratio. For a given source symbol it should return only the associated target-symbols and nothing more. Further it must allow the integration of context dependencies in the search process.

Data Representation

In the study of neural networks, one of the most important question to answer is: 'How to represent the data?'. The properties of the chosen encoding-scheme will determine the behaviour of the network to a large extent. In this case the options are restricted. There are some methods that will certainly not work, such as the often used n-gram method: shifting a window of a certain size over the words and vectorising the n-grams. Advantages of this method would be that it's easy to implement, it would give us the desired context dependencies and it would make it possible to code individual letters, rather then symbols, there by reducing the needed coding space.

However this method will not work, as the following example will show. Let us presume the following OCR-error: _formations_→_forinations_. Using a window-size of 5, the produced set of 5-grams would be : {_form→_fori, forma→forin, ormat→orina, rmati→rinat, matio→inati, ation→natio,tions→ation, ions_→tions}. Now the Network would, unjustly, be led to "belief" in the existence of some non-existent transitions. This problem is caused by the fact that the input and output word may vary in length. Using the notion of "symbols" instead of characters, and employing a string matching algorithm to spot the errors, seems the most natural way around this problem.

If a symbol representation is chosen, the question how to vectorise these symbols is still not answered. There are two possibilities: distributed or local representation. A distributed representation scheme for the output symbols, however, will not work. Since one must be able to retrieve a *set* of output-symbols for a given input-output symbol, a distributed representation would overlay the vectors of the output symbols. In this case it would not be possible to reconstruct the constituting vectors.

When the alternative, local representation, is used, the vectors of the output symbol would still be overlaid. But since there is no overlap between the coding for different output-symbols the desired result can be obtained. The output vectors will represent the output symbols that were seen during training. The other symbols, context and input, can be coded local or distributed.

For reasons of simplicity and uniformity we choose local representations. The representation used in our experiments consisted of an left-context part, which was used to hold the letter left to the input symbol, an input part, a right-context part and a output part.

Data Reduction

To make the task more simple for the neural network and at the same time speed-up the learning process we decided to reduce the data set. By using the data analysis describe above one can conclude that many symbols will hardly ever be used, and if used would probably be "forgotten" by the network anyway .

So we decided to exclude all input- or output-symbols that occurred only once from our representation scheme thus dramatically reducing the dimension size. The errors in which these symbol were used were eliminated from the training-set as well. Further reductions was done by excluding all transitions that had input-symbols that would never be searched, e.g. the symbols "." or "[". After this reduction we had an total of 155 symbols of the original 367 throwing away 16% of all errors. After this reduction the total length of the vectors was 201.

Back Propagation

Since the model behind OCR-errors can be seen as a mapping function, the most logical options for the choice of the network seems to be Back Propagation. Back Propagation is a supervised neural network. In the training phase input and output patterns are presented together. The network then learns to map the input to the output. This is done by adjusting the weights according to the error made.

Back Propagation has some drawbacks. Convergence is not guaranteed. The network might not converge at all, or end up in a local minimum (find a sub-optimal solution) but this is the case with all neural network models. Another drawback is that Back Propagation is very slow.

In the case at hand, Back Propagation would be a overly sophisticated method since Back Propagation is specialised for problems which are not linearly separable and therefore very complex. In this case, the coding of the OCR-errors was set-up in such a manner that the problem becomes linearly separable. Notice that the input vector are linearly separable, for every transition there is always only one node that is turned on. Some preliminary testing was done with Back Propagation, which showed that the high-dimensionality of the vectors made Back Propagation very slow and badly converging. Finally a good âlternative was found that was both simple and fast and well suited for the job; the Kohonen Feature Line. This architecture of this modified Kohonen network will be described below.

Kohonen Feature Line

The Network that was chosen implemented a Kohonen Feature Map with some adjustments. Kohonen Feature Maps are unsupervised learning networks, i.e. during training only one pattern is presented to the net, instead of an input-output pair as in the supervised-learning case. So, how can one train a input-output mapping to an unsupervised neural network? This is very simple. Just train the network with concatenated input and output vectors as input patterns. The network will then store these large vectors. To find the output for a certain input we first determine the node that represents the input vector best. Then we inspect the output part of this node, i.e. the dimensions of its weight-vectors that are used to encode the output.

To get this functionality, the net needs to be organised on input symbols. Every winning node of the network should be specialised for an specific input symbol and just that symbol. To obtain this result, the winner was determined during the learning phase on only the input symbol dimensions of the vectors. If the net would have been trained with the complete vectors, transitions that share common output symbols, rather then common input symbols, would win for the same node, thus messing up the organisation.

As in many cases, the topology of the Kohonen Feature Map is only used in the training process, not used in the fuzzy search process. The topology even introduced the problem that output symbols of neighbouring nodes were wrongly interpolated on some nodes. By making the Kohonen Map one-dimensional (Kohonen Line) this problem was dramatically reduced. Since a node in an one-dimensional map has less neighbours than in a higher dimensional map, unwanted neighbourhood effects are diminished. Further reduction of this problem was reached by setting the initial- and final-neighbourhood size parameters (σ-max. and σ-min.) in such a way that most of the learning time was spent to the fine tuning of single nodes. Good values for these parameters, that determine the area of the map that may be adjusted during learning, were 3 for σ-max. and 0.01 for σ-min.

The adjustments described above make the Kohonen Network resemble the Competitive Learning model closely, since Competitive learning is the same as a Kohonen Feature map without topology. Indeed a Competitive Learning network would be an alternative. The advantage of the Kohonen Feature Map, however is that there is no initialisation problem as there would be with a Competitive Learning Network.

Competitive Learning Networks have the danger of forming dead nodes and super-winners. The first are nodes that don't represent anything, the latter nodes that represent many patterns.

The way around this problem would be to initialise the weight vectors with the input symbols. This would mean that the capability of the network to forget unlikely transition is lost, the network is forced to code everything.

Therefore, the Kohonen Feature Line seems the best option. The size of the network was varied during some training runs and it was found that a network of 200 nodes organised well and hardly contained any useless nodes: nodes that never would become a winner.

Training the Network

A network would be well-organised after 3000 learning step The networks that were used for testing were trained 1000 step. The figure below shows the global organisation of small part of the network. The letters indicate the output symbols, the numbers the weight strength. Table 8.1 shows the topological organisation that is typical for the Kohonen feature map.

al:0.99	al:1.0	al:0.34,r:0.65	r:10
nodes →			

TABLE 8.1: GLOBAL ORGANISATION OF THE NET

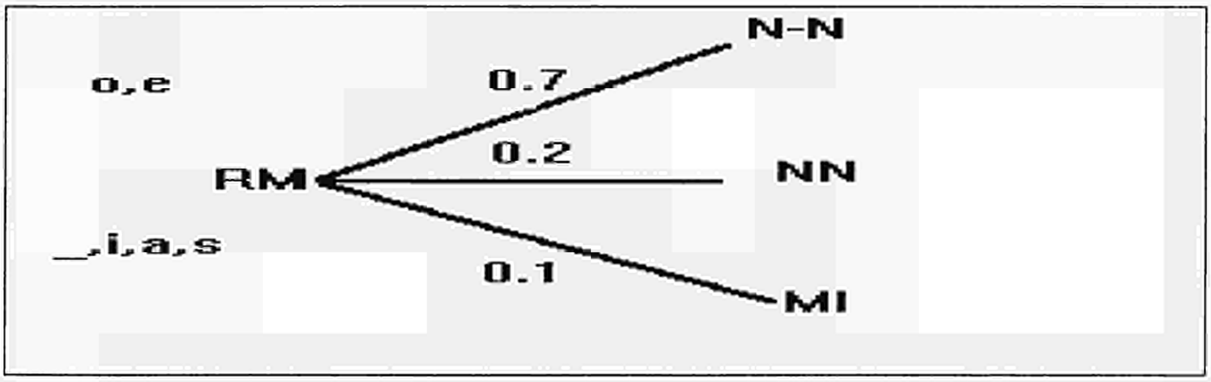


FIGURE 8.5: THE ORGANISATION WITHIN A NODE.

The organisation within a node is shown in Figure 8.5, and this is indeed the desired organisation. The information contained in a node enabled us to retrieve for a given input symbol (rm) the set of associated output symbols (n-n,nn and mi) and their likelihood by multiplying the weight strengths with the a-priori probabilities that were kept for the input symbol. Having to store the a-priori probabilities outside the network is unavoidable. The only alternative would be to train the network with all the text and not only the errors. However, networks are good noise-suppressers and using all the text would mean that the errors would simply become noise.

8.6 Results and Comparison

The following three methods were tested: Confusion matrix-, neural network- and wild card search. All tests were performed on some small, four to seven letter-sized, high-frequent words. Testing was done on a text-database that contained more than 30 megabytes of computer magazines.

For words smaller than four none of the methods seemed to work anymore. For words longer than seven the wild card search is a very good method and doesn't have the precision problems we were trying to solve. Recall for both statistical methods was measured relative to the wild card search with fuzzy degree one. I.e. what percentage of words, that the wild card search found, was found.

For both the neural network and the confusion matrix a rank threshold had to be set depending on the size of the word. Meaning that for a word of four letters, the first ten alternatives were considered serious candidates. The first 100 for words of size five, 150 for size six and 500 for words of length seven.

Both the confusion matrix and the neural network performed very well for short words. The set of words found with the neural network was in all cases a subset of those the confusion matrix found, which is of course very logical. Sometimes the words that were not found using the network were not alternatives for the wanted word. This explains the higher precision the network seems to have. However the recall of the network was considerable lower than that of the confusion matrix. On average the network scored an recall of approximately 75% and while the confusion matrix scored approximately. 90%. Figure 8.6, printed below, shows the precision values for the three methods.

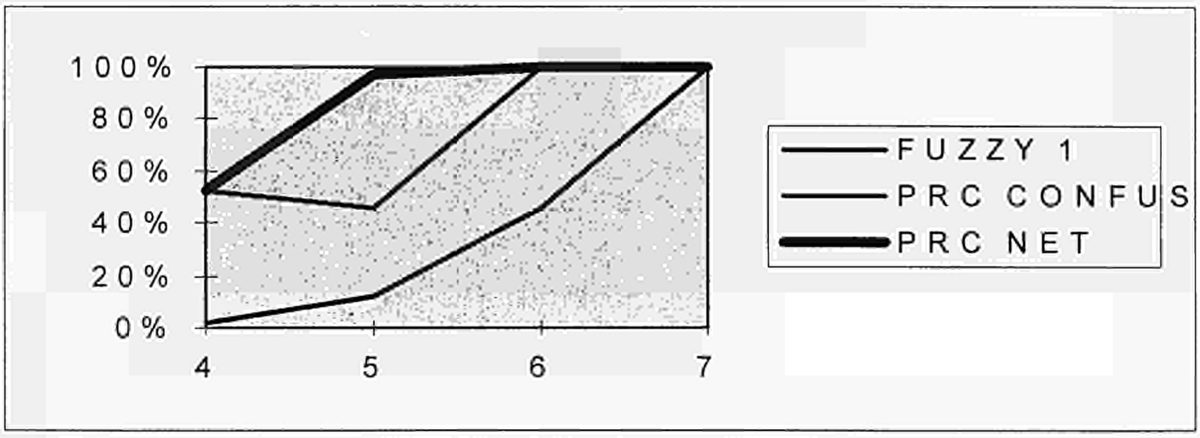


FIGURE 8.6: PRECISION FOR THE THREE METHODS.

Context Dependency

The results for the neural network that have been presented were obtained without the use of the context dependencies. Some testing was done using these dependencies to see if they were useful in the retrieval process. This led to no improvement.

There might me a number of reasons for this negative result.

- First of all not enough data was used to be able to find the context dependencies. In order to say something significant about context the data set that is needed should be orders of magnitudes larger. And we were not able to get hold of such a data set.
- Secondly, the question if OCR-errors are context dependent is still unanswered. If this is not the case, then this might explain the negative result.
- Thirdly, finding a good way to incorporate context in the search process is very hard. There are two extreme position one can take here. One can take these dependencies very serious, which would result in lowered recall especially when using such an small data set. Another extreme position is to take context not serious at all, but then they are useless. Finding the right position between those extremes is difficult.
- Finally manual inspection of the context-part of the nodes in the net showed that they tend to reflect a-priori letter frequencies rather than anything else This seem to suggest that OCR-errors are not context dependent.

8.7 Conclusions on Neural Networks for Fuzzy Searching

The following conclusions can be given:

- Using a statistical model to restrict the interpretation of the joker in a wild card search seems like a very good idea, especially for small words. A significant improvement in precision can be reached there. For longer words a wild card search is a very good method and should be used. Using a model that specialises for OCR-errors has the disadvantage that semantically related words are no longer found, although this can be solved by adding some stemming algorithm to these methods.

- Since finding OCR-errors is a problem, with a rather simple underlying statistical model, both methods seem to behave in a similar way. In a real application the confusion matrix should be preferred, since it is faster, easier to implement and to maintain. However the "forgetting" property neural network had a positive effect on precision.

- Whether OCR-errors are really context sensitive, and if adding context would improve retrieval remains an open question.

9 Clustering Prototype

-- *Jakub Zavrel*

Self-organising neural networks have been suggested for clustering and browsing document collections. In this chapter it is argued that growing cell neural networks proposed by Fritzke offer solutions for some of the problems that have been identified so far, namely cluster separability and scalability. A number of experiments is described on artificial and real-world data sets. The experiments confirm the expectations, and show that neural networks perform at a level comparable to traditional methods. However, they do not offer a free lunch.

9.1 Introduction

Library catalogues are stored in computerised databases. Typically, traditional Database Management Systems (DBMS) are still the only tools used for this purpose. The information associated with a certain item in the collection is stored in records which are further structured into fields, e.g. author, title, ISBN number, catalogue code, etc. The information is made accessible by Boolean search techniques. This retrieval method is particularly well suited for the traditional task of finding the catalogue code, given the exact bibliographic description. When one has less definite descriptors to start a search with it is often hard to access the collection in an effective way.

Full text

At present, technology allows more and more libraries to enrich their catalogues with large keyword sets, abstracts, or even full text electronic versions of a text. While these could obviously be of great use in the structuring of the collection and the improvement of accessibility, the traditional DBMS is not well equipped to deal with such natural language information. Full text information retrieval (IR) capabilities will have to be incorporated into the On-line Public Access Catalogues (OPAC's) of the future.

Explorative search

Full text IR responds to the user's query by returning those documents that are considered most relevant to the query. However, often the casual searcher does not know what exactly to look for, or does not know how to formulate the right query. The contents of the catalogue are like a deep dark sea to the user. A query is like a fishing net. A fishing net however, is a poor instrument when one wants to explore where the fish are, if the fish are near but not in the net.

A very useful strategy for explorative search in a library is often to skim for relevant titles in the physical bookshelves where a skilled librarian has arranged the books in some practical grouping. The present study aims to investigate whether the clustering of documents with neural networks can be of use for tools that allow for explorative search and browsing in a large document collection from behind a terminal.

Semantic road maps

[Doyle, 1961] already foresaw the need for such tools and argued for computer aided construction of what he called : "Semantic Road Maps for Literature Searchers". The argument was that "to increase the mental contact between the searcher and the information store" one needs to give the searcher a visualisation of the collection contents so that he or she can "home in on the relevant documents just as one would in a supermarket". The distance in the visualisation between certain items should reflect their semantic relationship. The method Doyle suggested is the automated analysis of co-occurrences between words within and between documents in the collection.

Visualisation with neural networks

These ideas have recently reappeared on some scale in IR, due to the increase of available computing capacity and the advances in visualisation technology (see e.g. [Hemmje et al., 1994]). As has been noted by several authors ([Lin et al., 1991], [Scholtes, 1993]), there is quite a striking resemblance between the proposed functionality of Doyle's semantic road maps and the properties of topologically organised neural networks (e.g. [Kohonen, 1990b]). These neural networks perform unsupervised clustering and topological ordering of input patterns onto a neural map. Lin et al. and Scholtes have experimented with the application of Kohonen maps in IR and encountered both successes and problems. Four of the main problems that have been observed are: scalability, cluster separation, unreliability of convergence, and evaluation.

Problems

- Scalability is the problem that when one demonstrates the applicability of a neural network to a toy version of a task, it does not automatically mean that the solution will scale well to a real world version of that same task.

- Cluster separation can be described as follows. The visualisation of a document collection on a two dimensional screen is a drastic dimension reduction: a very high dimensional 'semantic space' is transformed to the two (or three) dimensions of a screen. In the case of the Kohonen network this can lead to the fact that two documents are far away from each other in the semantic space, but end up very close together in the network. This makes the distance measure in the network unreliable.
- A problem which is of importance for practical applicability is that the self-organisation process of the network may not converge reliably into the desired topological organisation.
- Finally, a methodological problem is that of evaluation. How to evaluate new ideas that are aimed at "increasing the mental contact of the user with the information store"? If one could make the testing of user satisfaction not too subjective, certainly it would still remain very time-consuming. An alternative approach, which has been used in the present work, is to compare the retrieval effectiveness[9] of the new tool to traditional methods.

Clustering in information retrieval

After all, in traditional IR clustering has been an important method [Salton, 1989], [Willett, 1988]. Similar documents in a collection are organised into smaller groups. It is used to increase both efficiency and effectiveness.

The amount of documents in a collection is often very large. In a distance based retrieval model such as the vector space model, where relevance is taken to be proportional to the distance between the term vector of the query and the term vector of the document, the comparison of the query to every single document would be too costly. To reduce searching time the whole collection is divided into clusters beforehand. At the moment of searching the query needs to be compared only to the clusters (which are fewer in number) and then to the few remaining documents that are in selected clusters.

Although one might suspect that clustering should be less effective than a complete search, this is often not the case. It has been found that for some collections [Willett, 1988] clustering can improve effectiveness. This is the case when the Cluster Hypothesis holds of a collection:

[9] Effectiveness is usually measured in terms of recall and precision, or some combination of these. Recall is the percentage of all relevant documents that have actually been retrieved. Precision is the percentage of the set of retrieved documents that is deemed to be relevant. Increasing recall damages precision, and vice versa.

"Documents which are similar to each other may be expected to be relevant to the same queries" [Van Rijsbergen, 1979]

Cluster based retrieval makes the recall for a given query higher, by retrieving documents that do not match the query very well, but that resemble those that *do* match the query.

Overview of this chapter

The work that is described in this chapter aims to build upon the earlier work in this direction and to suggest solutions for the problems described above by making use of a new variant of the Kohonen network called Growing Cell Structures [Fritzke, 1993].

In the following section traditional clustering methods are described, in order to provide a background and a rationale for performance comparison of the new methods. The next discusses the neural methods and the problems and advantages that have so far been observed in their application in IR. The third discusses the new neural methods that are supposed to be more suitable for our purposes. Section four reports experiments with the new neural methods on artificial data. In section five we will discuss the issue of scalability in more detail. Sections six makes an analysis of the data set that will be used in our experiments, and reviews some issues that are related to the representation of documents for clustering. Section seven describes the experiments in clustering bibliographic records and the evaluation of the results. Finally, a discussion of the neural network approach and a conclusion on the utility of neural networks for clustering documents follows.

9.2 Traditional clustering methodologies

There exists a large amount of clustering algorithms from the field of statistics. In choosing the appropriate algorithm one has to make a trade-off between effectiveness, i.e. the quality of the resulting clusters, and computational efficiency. The most effective algorithms, hierarchic agglomerative methods, require the computation of the full inter-document similarity matrix. Clusters are then formed by some criterion, for which the matrix has to be searched repeatedly. The complexity is $O(n^2)$ up to $O(n^5)$ in time, and $O(n^2)$ in memory space. For very large document collections this is too expensive. The main practical advantage of these methods is that they are well understood, have clearly defined goals (e.g. minimise inter-cluster variance, maximise extra-cluster variance), and there is a large body of both theoretical and applied research literature available.

The alternative is to use heuristic algorithms, which can cluster a collection in one pass by dividing it into partitions. However, these are not always reliable. The final results can be very sensitive to initial parameters and to order of presentation. In tests they have been found to be ineffective for information retrieval [Willett, 1988].

Both hierarchic and partitioned cluster structures are not very amenable to visualisation, because there is no simple projection of the clusters on the two dimensions of a computer screen. Are neural networks a remedy?

9.3 Self-organising neural networks for clustering

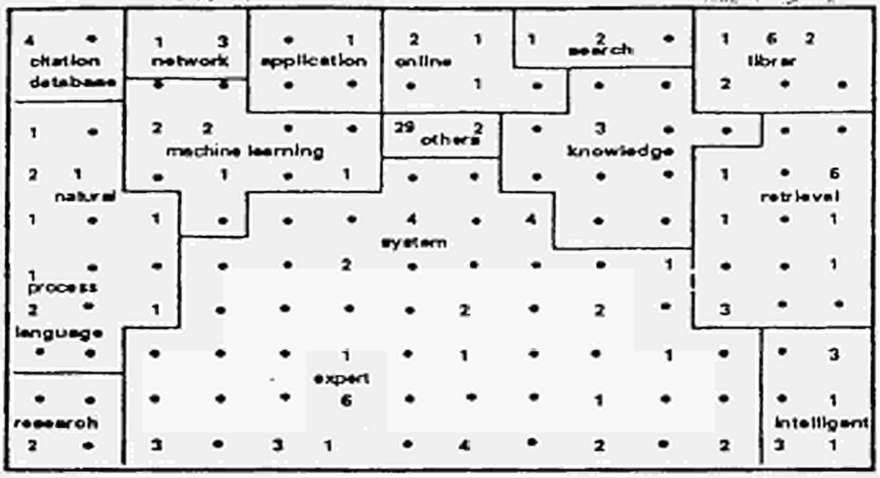
In this section we will describe some of the previous work that has led us in the present direction. Special emphasis will be put on the problems that have been encountered and on possible solutions.

Intuitive correspondence

One of the first attempts to use self-organising neural networks for clustering and or interface design is described in [Lin et al., 1991]. Lin *et al.* noted that :

> *"Emphasis on frequencies and distributions of underlying input data, understanding of the computer's role in producing an associative map similar to the feature map in the brain, and projection of a high dimensional space to a two dimensional map are, in fact, the three most distinguishing characteristics of Kohonen's feature map".*

Inspired by this intuitive correspondence, Lin *et al.* have implemented a prototype system in which 140 documents, each represented by 25 terms taken from the title, were presented for training to a Kohonen map of modest size[10].



A self-organizing semantic map of AI literature. 140 documents from LISA database are used as input to produce the map. The areas on the map are automatically generated, their relative positions, neighbors, and sizes are determined by the input data. The numbers on the map represent the number of documents mapped to each node.

FIGURE 9.1: NEURAL USER INTERFACE (REPRINTED FROM [FROM LIN ET AL. 1991])

The resulting map (see figure above) can serve as an interface for retrieval. It shows the following appealing features:

- *"It maps a high dimensional document space to a two-dimensional map while maintaining as faithfully as possible document inter-relationships.*
- *It visualises the underlying structure of a document space by :howing geographic areas of major concepts in the document space and the document distribution over the concept areas. Properties related to geographic concepts such as areas, locations, sizes and neighbours all reflect the statistical patterns of the document space.*
- *It results in an indexing scheme (weights) that shows the potential to compensate for incompleteness of a simple document representation. (title word indexing)."* [Lin et al. 1991]

[10] A rectangular grid of 14 x 10 nodes.

Although it was quite successful as a first attempt in this direction, it is clear that the scale of the task that has been tackled here is not at all similar in scale to a real retrieval environment. The question of evaluation is also deferred, by suggesting assessment of user satisfaction and comparison to a human-generated semantic map for this means. It is true that these are important measures, but it is hardly feasible at all to perform them on some reasonable scale with a degree of objectivity.

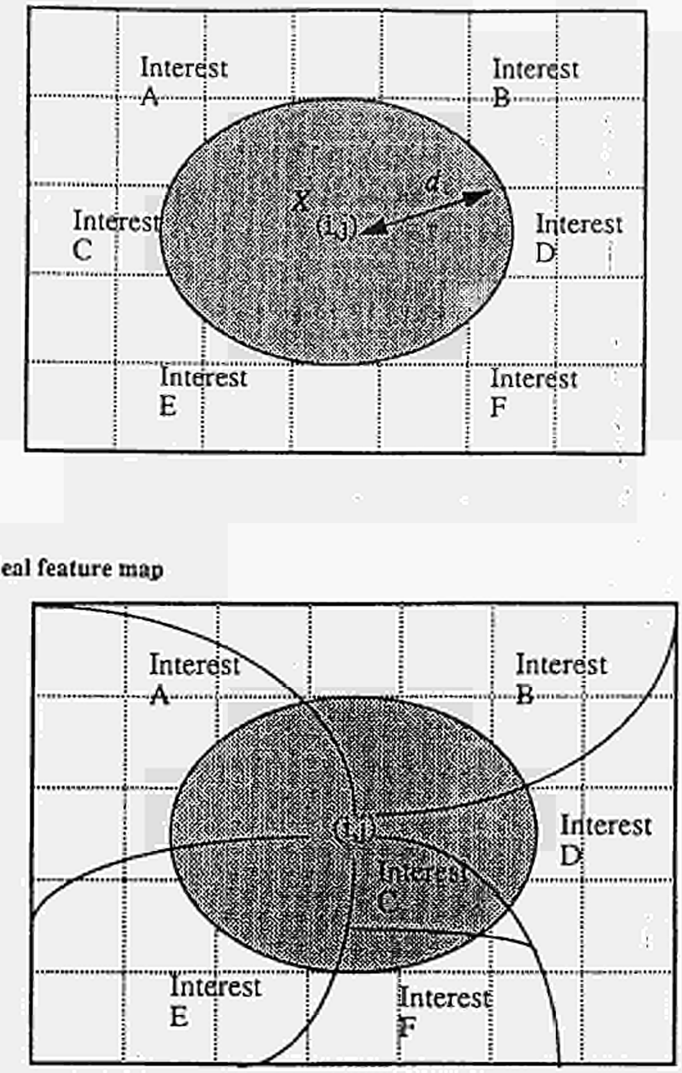
Identification of problems

[Scholtes, 1993] has also performed experiments in this direction with the construction of the "Neural Interest Map". This map was also trained with patterns representing a selection of terms from documents. Scholtes increased the scale of the representation of the documents considerably, clustering on 500 term vectors. This is important in moving towards clustering on full text of e.g. abstracts. We will return to the question of scale in section 3, where we will discuss the representation of documents.

Again, the visualisation of the organisation of the documents is very appealing, but a number of questions still remains. Is this method further scalable? And, most important, is it useful for information retrieval? [Scholtes, 1993] and the *Part 2 of this report: State-of-the-Art* both draw pessimistic conclusions about these questions, and observe the following two fundamental shortcomings of clustering high dimensional data with Kohonen networks.

Even when clustering a relatively small number of documents (50) Scholtes has found that the convergence of the topological map with the standard Kohonen algorithm takes a very long time and easily gets stuck in non-optimal local minima.

Another setback that Scholtes describes results from the uniform structure of the Kohonen map. The map consists of a rectangular grid of nodes with a fixed topology. As nodes in the map come to represent documents after adaptation, one expects that distance in the map reflects proximity of documents in the semantic space. The document space however, does not show a uniform distribution and is not inherently two dimensional. The Kohonen algorithm tends to spread out input patterns equiprobably over the map. A result of this is that the distance measure in the map becomes unreliable for browsing purposes. This is illustrated in figures below. In an ideal feature map (Figure 9.2a), if a query vector X is mapped onto node (i,j), then all nodes within distance *d* are supposed to be related (dark circle). In practice a derived map tends to be more like Figure 9.2b. If a query vector X is mapped onto node (i,j) here, it is positioned at the borders of several clusters. These clusters are not necessarily neighbours because they are related, but because they are forced to connect due to the dimension reduction properties of the map. Selection of documents within the distance *d* on the map would result in poor precision and recall[11].



a. An ideal feature map

b. A feature map as they occur in the real world

FIGURE 9.2: (REPRINTED FROM [SCHOLTES, 1993]).

[11] One might also take cluster boundaries into account in the retrieval process by looking at the distance of nodes in the document space. A high distance is suggestive of a cluster boundary. The criterion for what makes a distance high however, is very dependent upon the collection of documents, and therefore hard to determine. Another option is to compute so called tension regions [Henseler 1993]. We have not yet explored this option in depth.

A number of variants on the Kohonen map has been proposed that are meant to deal with these problems (Fritzke 1993, Blackmore & Miikkulainen 1993). The basic idea is that one can start with a very small network, adapt it to the pattern space, grow the network in regions where this is needed, and prune the network of spurious nodes. These networks are much less likely to get stuck in local minima, because they start out ordered and do not get tangled easily. They also adapt their structure to the structure of the task, so that e.g. cluster boundaries are reflected in it. They will be discussed in section 3.

Effective neural clustering

Another interesting approach towards neural clustering is that of MacLeod & Robertson (1991). They have designed a neural clustering algorithm based on ART networks (Carpenter & Grossberg 1988). This type of networks does not exhibit topographic mapping, but only adaptive categorisation. This indicates that it could not be used in its original form for a browsing interface. However, MacLeod & Robertson report promising results. Their algorithm has a computational complexity that lies between that of heuristic one-pass methods and hierarchic agglomerative methods. The effectiveness however, as measured on two standard test collections, is comparable to that of hierarchical clustering.

Desiderata

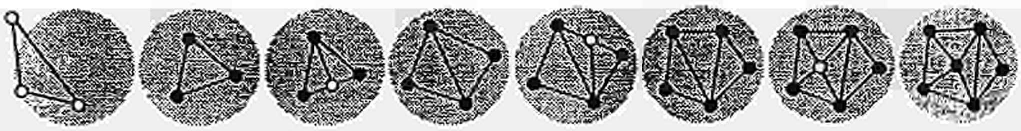
For clustering purposes it would be desirable if the structure of a neural network would adapt to the structure of the input space. The standard Kohonen algorithm has a fixed grid structure. A number of proposals have been made to grow the network structure in response to the characteristics of the input. This facilitates convergence and enables us to visualise cluster boundaries.

9.4 Growing neural networks for clustering

Fritzke's growing cell structures

The network proposed by [Fritzke, 1991a, 1991b, 1993] has a structure which consists of k-dimensional simplexes[12]. The vertices of each simplex are nodes which have a position in the input space denoted by their associated weight vector. The edges stand for neighbourhood relations. The structure starts out with just one simplex, with randomly initialised weights. Patterns are presented and after each presentation the weights of the BMU (best matching unit) and its neighbours are updated. This is continued for some time so that the structure is topologically well-organised. Unlike in the Kohonen algorithm the adaptation gain is constant through time.

To better adapt the structure to the pattern space, the network is slowly grown from its small initial configuration. In terms of clustering, the network starts out by dividing the pattern space in a small number of rough clusters, and inserts new clusters in places where this is needed. The 'need' is estimated from the patterns that have been encountered already. Each node **c** has a signal counter variable $\mathbf{T_c}$ which is increased by a fixed amount if that node becomes the BMU. After some interval of adaptation steps, the node with the largest relative signal frequency[13] $\mathbf{h_c}$ is singled out as the place to grow the network (see Figure 9.3). This node is referred to as the black sheep.



Development of a cell structure for a circular probability distribution

FIGURE 9.3: (REPRINTED FROM [FRITZKE 1993]).

[12] For k=1 these are lines, k=2 triangles, k=3 tetrahedrons, k>3 hypertetrahedrons. In the simulations in this paper we have restricted ourselves to k=2. This seems an appropriate choice for visualisation.

[13] $\mathbf{h_c}$ is the signal counter $\mathbf{T_c}$ divided by the sum of the signal counters of al nodes in the structure.

The largest $\mathbf{h}_c$ indicates that the black sheep node covers the largest portion of the input patterns. The edge between the black sheep and its furthest neighbour is the place where a new node is inserted[14]. The edges are then updated so that the structure consists of simplexes only. The network approaches a state where the density of the nodes gives a good estimate of the probability density of the input space. Cells that are positioned in regions with low probability tend cover a large part of the input space but are seldomly BMU (low signal frequency). They can effectively be pruned out of the structure. Boundaries between clusters are therefore reflected in the network. The difference between the Kohonen map and the growing cell structure on a 4-cluster problem can be seen in Figure 9.4

[14] It should be noted that nodes can be inserted anywhere in the structure, not only at the borders. The same can be said for deletions.

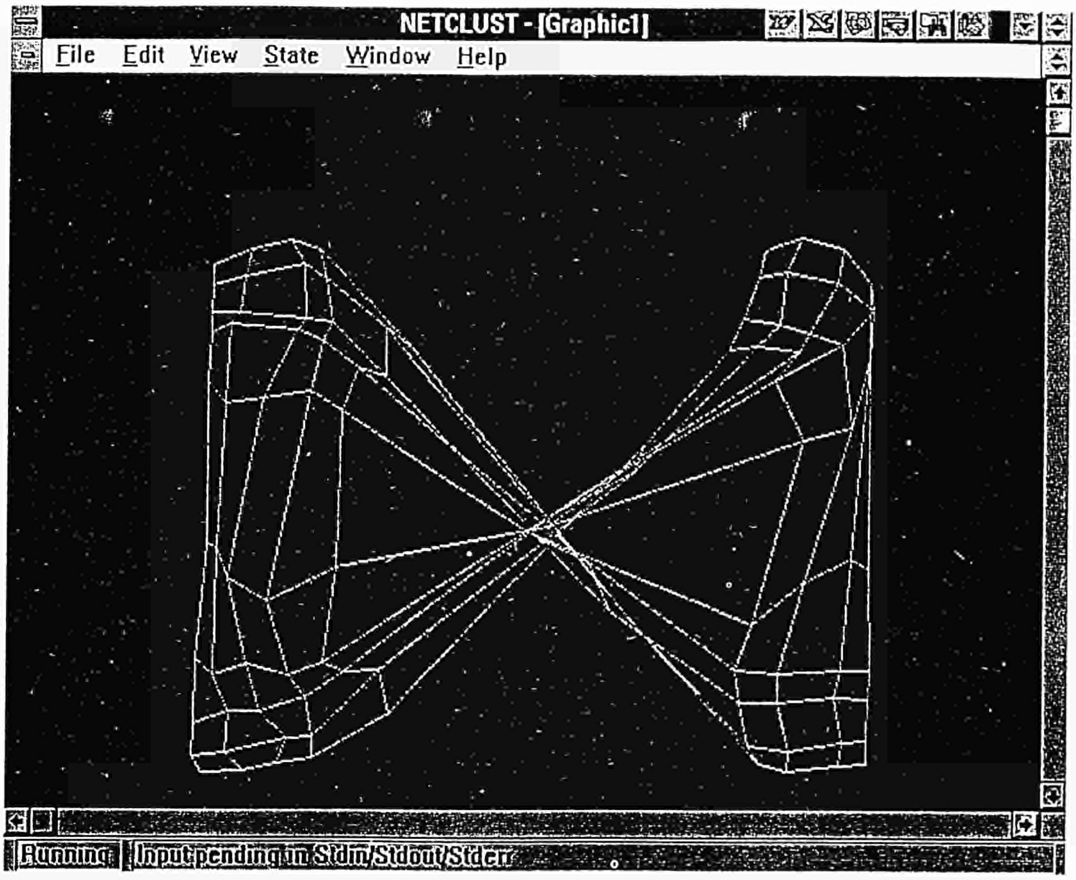


FIGURE 9.4: KOHONEN NETWORK GETS TANGLED EASILY AND DOES NOT SEPARATE CLUSTERS WELL.

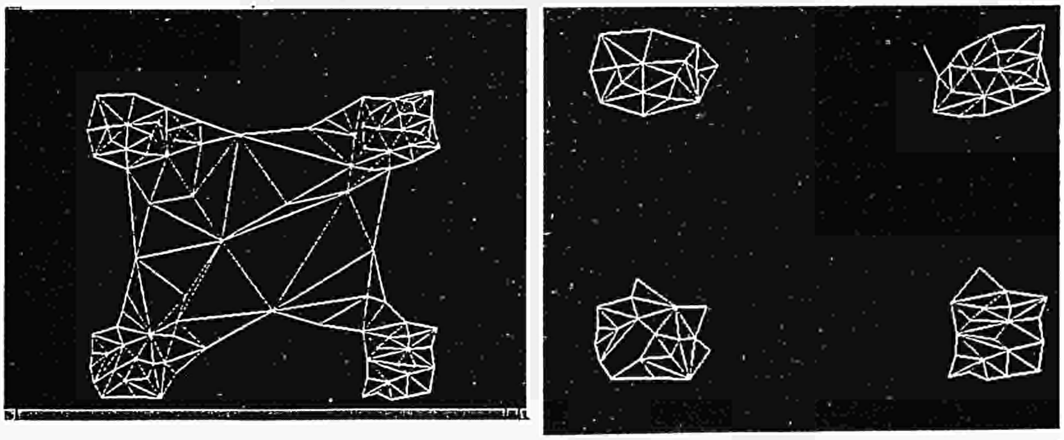


FIGURE 9.5: FRITZKE NETWORK EASILY SEPARATES FOUR CLUSTERS IN TWO DIMENSIONS (BEFORE AND AFTER PRUNING).

In the case of a two dimensional network structure (triangles) the network can be used for visualisation. However, the fact that the structure of triangles can in principle always be drawn in two dimensions, does not necessarily mean that this is always easy. If the dimension of the weights is higher than 2 or 3 (it is very high in our application domain), the network can no longer be drawn by projection on the input space. Sophisticated layout algorithms are then needed to produce a nice drawing fast[15].

Gridnet

As a reaction to the problem of visualisation another approach was taken by [Blackmore & Miikkulainen, 1993]. Their network is grown on a fixed rectangular grid. We will further refer to it as Gridnet. The network starts out as four connected nodes (see Figure 9.6), with randomly initialised weights.

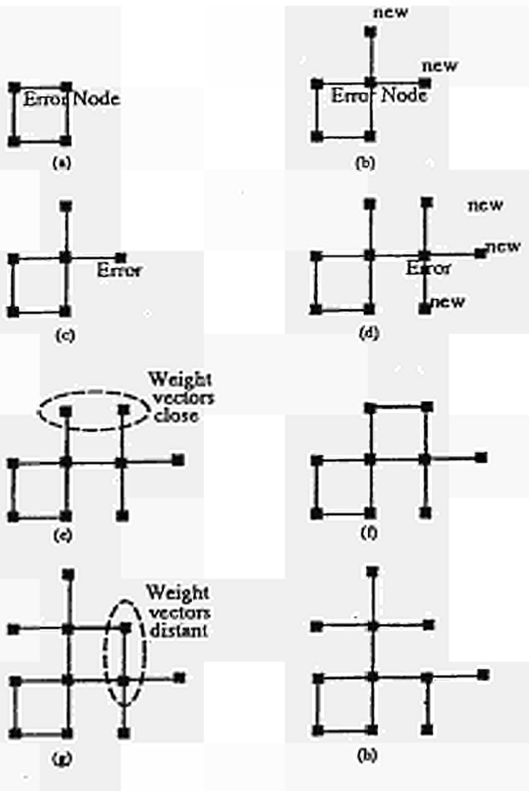


FIGURE 9.6: THE GROWTH PROCESS OF GRIDNET (REPRINTED FROM [BLACKMORE & MIIKKULAINEN, 1993]).

[15] Fritzke proposes a method which uses a physical force metaphor. The nodes can attract and repel each other. [Blackmore & Miikkulainen, 1993] claim that this method does not always result in correct drawings.

Patterns are presented and the weights are adapted the Kohonen learning rule. For each node an error variable is kept. If a node becomes BMU, its error variable is increased, with the squared distance between the weight of the node and the input vector. After a certain amount of time the node with the highest error variable is selected for growth. New cells are placed in all neighbouring grid locations. Nodes are never deleted, but the neighbourhood connections between cells can be deleted to indicate cluster borders. For this purpose connections are periodically subjected to a distance threshold. If two nodes that occupy neighbouring locations on the grid are closer together in the input space than a *connect* threshold a connection is made between them. If neighbouring cells are too far apart, the connection between them is deleted. Two points should be noted. The network can only grow outward at the boundaries, and the distance criterion for (dis)connection is problem dependent. Experiments have show that these properties make Gridnet perform worse in cluster separation than Fritzke's network. In the next section a comparison of Kohonen, Fritzke and Gridnet networks is made on a artificial clustering and retrieval task similar to our final bibliographic task.

9.5 An artificial retrieval task

The success or failure of any clustering method is extremely dependent on the characteristics of the document collection. It is easy to imagine that a collection which simply does not contain clear clusters will not effectively be searched by cluster based retrieval[16]. To abstract away from the characteristics of any particular collection and to ascertain whether the growing cell networks really are suitable for the intended use, a first test was performed on a artificially generated data set with characteristics similar to those of an ideal document collection. On this data set Fritzke and Kohonen networks, and Gridnet were tested.

The artificial data

500 'document' vectors of 100 dimensions were generated by making 25 random cluster prototypes and generating 20 permutations of each prototype. The prototypes were sparse, like the real documents, and had about 10% of the terms set to 1. A permutation had 5% of its dimensions flipped (from 1 to 0 or vice versa). The resulting data set consists of 25 non-overlapping clusters.

Clustering

These patterns were presented to the three different types of network. After the adaptation was stopped (5000 pattern presentations) all the patterns were presented to the network once again with learning shut off. This is the cluster compilation phase. For each pattern the node that responded best to it was taken to represent it. After cluster compilation each node in the network has a list of zero or more patterns associated with it.

Retrieval

In the retrieval phase a query is entered and presented to the network in the same way as the documents. The node that responds best holds the cluster to be retrieved. Often the clusters are more distributed over the network. To retrieve more relevant documents one would browse the neighbouring nodes as well. Because for the artificial data set the relevance judgements are clear - every other pattern that was generated from the same prototype is taken to be relevant - recall and precision can be measured. This is done at two rates of recall.

[16] [Willett, 1988] describes a number of tests that can be performed on document collections to determine whether clustering is appropriate.

If only the list of the BMU is retrieved recall is low and precision high. If the neighbours are retrieved as well recall is higher but precision drops. I.e. precision drops if and only if the extension of the retrieved set to the neighbours crosses a cluster boundary. It should be expected that the growing cell networks have less problems with this. This can indeed been seen in figure below.

Comparison

The pictures in Figures 9.7, 9.8 and 9.9 below show the overall structure of the clusters on the map. The vectors from a cluster are categorised in each others neighbourhood. This is marked by border-lines in the pictures. The larger numbers give the label of the prototype from which the vectors in that region were generated. The smaller numbers are vector numbers in Figure 9.7 and 9.9, and unit numbers in the network depicted in Figure 9.8. The Kohonen map clearly shows that clusters are evenly distributed and borders between them are not delineated. The overall structure of the Fritzke network cannot be visualised as easily, the drawing was produced manually, but the separation of the clusters by the pruning heuristic is near perfect. Only in a few places has a cluster been split erroneously. The Gridnet, which is easy to visualise in the same way as the Kohonen map, does unfortunately not show this excellent cluster separation. The few clusters that have been able to grow at the border are clear, but the clusters which remain in the centre of the structure have not been separated by the pruning heuristic. The parameters of the pruning heuristic are very problem dependent in Gridnet, this in contrast to the Fritzke network.

FIGURE 9.7 - 9.9. CLUSTER STRUCTURE ON 25 CLUSTERS OF ARTIFICIALLY GENERATED DATA.

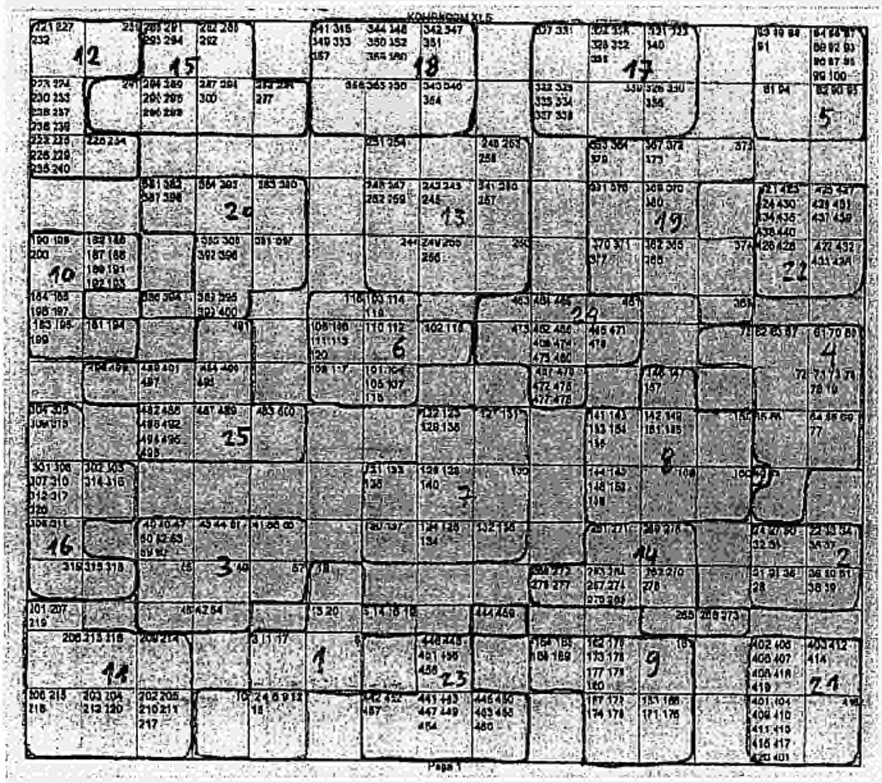


FIGURE 9.7: KOHONEN NETWORK

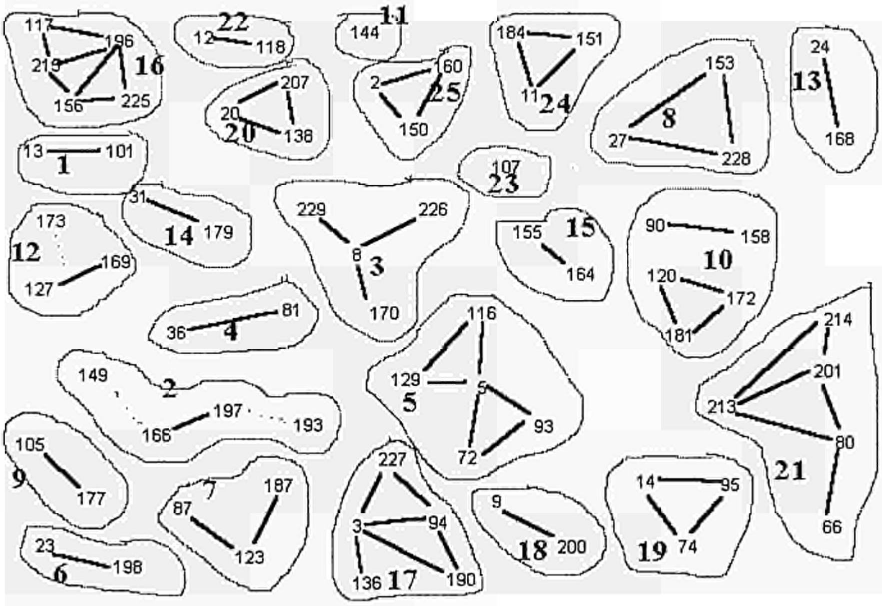


FIGURE 9.8: FRITZKE NETWORK.

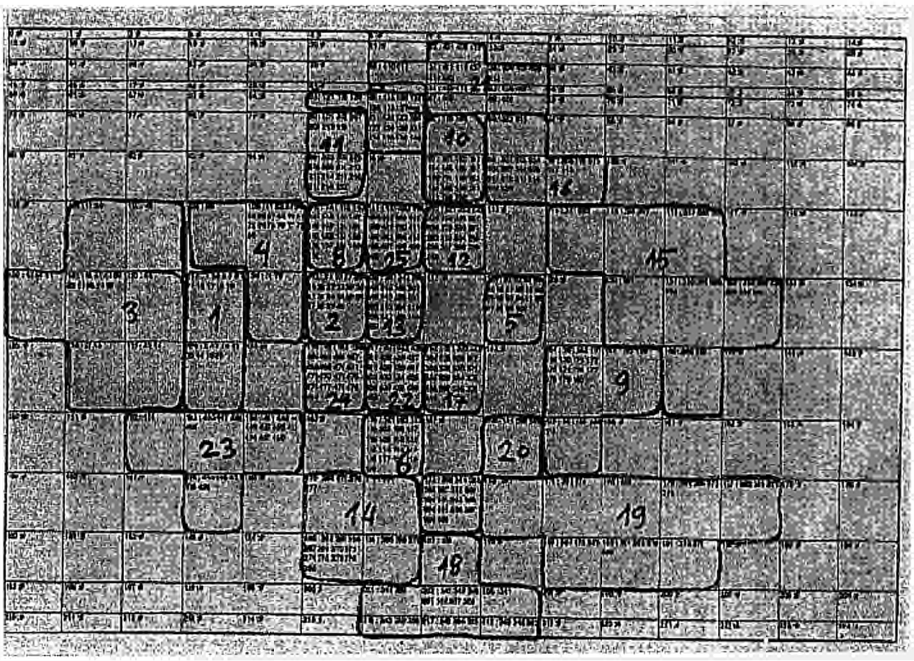


FIGURE 9.9: GRIDNET.

The visible properties of the clusters on the three different networks can be translated into recall and precision figures. These are shown in Table 9.1.

Network type	Size	Retrieval strategy	% Recall	% Precision
Kohonen	10x10	bmu	60	100
		bmu+neigh	99	72
Kohonen	15x15	bmu	22	100
		bmu+neigh	78	86
Fritzke	100	bmu	*56*	*100*
		bmu+neigh	*95*	*100*
Gridnet	100	bmu	66	100
		bmu+neigh	95	**43**

TABLE 9.1: RECALL AND PRECISION ON 25 CLUSTERS OF ARTIFICIALLY GENERATED DATA.

Drawing a conclusion from these tests on artificial data, we can conclude that Fritzke's growing cell network is superior for this task. Next it will be applied to real bibliographic records. But before that we will look into some issues of scale and data representation.

9.6 Issues of scalability

Size of the data

What does it mean for an algorithm to be scalable? Suppose that the qualitative problems of clustering with Kohonen networks are solved by Fritzke's innovations. Does this mean that the whole issue of neural networks being badly scaleable is solved? There are, in fact, two separate notions of scale in document clustering:

- There is the size of a document collection which is usually very large. This puts a heavy computational load on any clustering algorithm. This is not only so because of the number of documents that hás to be dealt with, but also because the number of clusters (of reasonable size for retrieval) that have to be distinguished grows with the size of the collection. The neural clustering methods do not require to store or search a document similarity matrix, so they can be expected to scale better than hierarchical clustering methods in this respect. The size and number of clusters is given by the number of nodes in the network, and in this respect neural networks are of course slowed down by the size of the collection.

- The second scale of a clustering task is the number of terms which are used to represent the documents. E.g. [Lin et al., 1991] used 25 terms, [Lelu, 1991] and [Scholtes, 1993] used 500, but a large Boolean index can have over 200,000 terms. And this is clearly out of reach of current computational power for vector-clustering. In the next chapter a number of proposals are made to make the number of terms manageable.

Complexity of the network

Each presentation of a pattern to the network consists of the comparison of an **n**-dimensional input vector with the weight vectors of **m** nodes. So the time for one adaptation step is O(**n**·**m**), where **n** is the number of terms, and **m** is the number of clusters (nodes). The required number of clusters is, as we have noted, proportional to the number of documents.

The number of adaptation steps needed is at least proportional to the number of documents, so that we have a time complexity of O(**n**·**m**·**s**), where **n**,**m** as above, and **s** is the size of the collection. As **m** is dependent on **s**, we end up with a total time complexity of O(**n**·$\mathbf{s}^2$), at the lower bound of the complexities of the hierarchic methods. The promise of parallel execution of NN brings this down to O(**n**·**s**). For the neural network we need to store only the weight vectors for each node, so that the space complexity is O(**n**·**m**).

The difference between neural network approaches and many other methods is that there is no clear indication of when the clustering process is finished. So it is unclear what the constant factor is in the computational complexity, or whether it is constant at all. It might very well be that the required training time to get a *good* clustering increases steeper than linear with the size of the collection. We have used simple heuristics to stop the process in our simulations in the previous chapter, but in principle one could improve the clusters *ad infinitum*. We are not sure yet whether this feature is an advantage or a disadvantage. This was investigated to some extent in our experiments, and more will be said about it in their discussion in section 9.9.

9.7 Data analysis and representation

It is necessary to determine whether the amount of terms in a typical document collection justifies the claim that neural networks are a feasible technology for full text clustering.

Statistics of a document collection

The data set which we initially had available consisted of a subset of 19,235 records from a Dialog CD-ROM containing the 1980-September 1991 Current Index to Journals in Education (CIJE) and Resources in Education (RIE). The records in this database consist of many fields with various properties. Fields like ISBN number or author are not really suited for clustering. A field like the title of the journal from which a document was taken is very telling of the content. Search on this field could however, hardly be better than Boolean search. The fields that were considered were: title, abstract, and descriptor keywords. Only the first two are genuine free text. Figure 9.10 below, which depicts the increase of the number of different terms as a function of the size of the collection (for our data set) can give us a good feeling for the issue of scalability in number of terms.

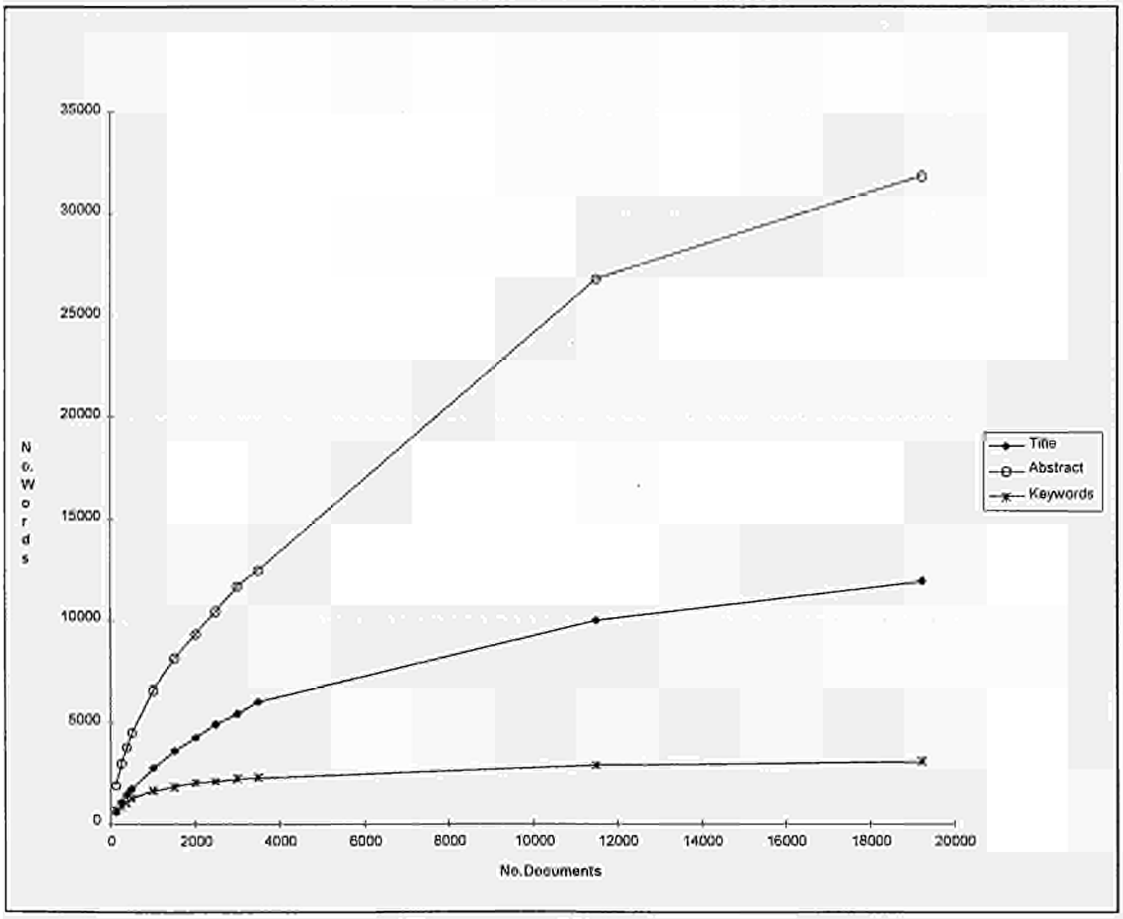


FIGURE 9.10: NUMBER OF WORDS GROWS WITH COLLECTION SIZE.

As one would expect, the total amount of distinct words grows to an asymptote as the collection size is increased. The difference between the fields is where this asymptote lies. For the most typical free text field, the abstract, one can see that the number of words can grow very large and does not level off very fast[17]. The same is true of the title field. Not only are the absolute numbers beyond the capabilities of the computing machinery which we had available, it is also likely that the steepness of the growth will cause a problem in the case of extension of a collection. The experiments that will be described below were conducted with the terms from the keyword-field. This field levelled off at about 2500 distinct terms.

Reducing the number of terms

For effective clustering the terms used have to be as semantically discriminative as possible. However, it would be problematic to select important index terms manually, especially because the importance of terms within a collection might change as the collection's content changes. We have considered several options to deal with this. For our simulations with this data set we have finally chosen not to cluster on full-text, but to select the 1000 most frequent descriptor keywords remaining after a stemming and stoplist pre-processing step. The other options considered were compression of the document vectors, and triplet coding. These latter two methods involved too many open ended research questions for the time available for this project, so they will be discussed only shortly here. After that some justification will be given for the chosen method.

Compression of the document vectors

One important observation about the number of terms is that the occurrence of a term in documents is far from independent from the occurrence of other terms. This is one of the reasons that clustering can be successful at all. It must be possiole to exploit this redundancy and compress the number of terms by using a more efficient information representation. Such a compression can be achieved by using a auto-encoding backpropagation network with less hidden nodes than input and output nodes. The use of backpropagation for this task was shown to be successful by [Cottrell et al., 1987]. [Baldi & Hornik, 1989] subsequently demonstrated that such a network learns to extract the n first principal components of the data, where n is the number of hidden units.

[17] Note that this increase would be even steeper if the data set had not been restricted in subject (education).

Principal Component Analysis

Instead of using a network one could also apply principal component analysis (PCA) to the full document-term matrix. PCA tries to select a new basis for a vector space by finding the directions of maximum variance in the data set. [Schütze, 1993] has shown that singular value decomposition, a related technique, can be very efficient for compressing natural language data. We will not go deeper into this issue here because we feel that the use of PCA in IR is a full scale research issue in its own right. Our research in this direction is still in progress, and preliminary experiments show that PCA may not only help clustering, but might very well be used for visualisation purposes by itself.

Random projection

Another possible method for reduction of the number of dimensions is described by [Ritter & Kohonen, 1989]. It exploits the sparseness of the document vectors - a typical document contains only a very small subset of all known terms - to justify a dimension reduction of the original documents space by a random projection onto a space with a much smaller number of dimensions. It can be proven that this random projection preserves the relationships in the original space faithfully with a very high probability.

Triplet coding

The number of terms grows rapidly in free text and is both topic and language dependent. It also seems attractive to try to represent the documents in a vocabulary independent way. One method to do this is to use triplet representations. Triplets are strings of three consecutive letters. The number of possible triplets is bounded at 27^3 (19,683), but only a much smaller number of them actually occurs in text. For best match retrieval this is a successful method [De Heer, 1982],][Teufel & Schmidt, 1988], [Scholtes, 1993], but to our knowledge this method has rarely been used for clustering yet.

The method of choice

To avoid unexpected problems due to the combination of two or more new methods, we ended up using a simple method to represent the documents and terms in the simulations. Each document is represented by a vector of as many dimensions as the total number of terms. If a certain term is present in the document, the corresponding dimension has value 1, otherwise it is 0. One could use a weighted value to emphasise the importance of each term. According to [Willett, 1988], however, this makes very little difference for clustering.

Because the program we used for the simulations was run on a common PC, the number of terms which could be handled in reasonable time was about 1000. The number of distinct terms in the keyword field of our database was 2411. We dropped terms so that the 1000 most frequent terms remained. The legitimacy of this move might be questioned on several grounds. Certainly it discards some information. The most frequent terms however, contribute most to the overall clustering. One could also see this as a transformation of the document set, to one with a smaller controlled vocabulary. All comparisons with other methods are based only on these 1000 terms. It should also be noted that the remaining terms, while less than half of the original number of terms, cover 96% of the total contents of the keyword field. A similar cut-off for the terms in the abstract field would leave us covering only 60% of the text, and setting the cut-off at 96% for the abstract field would still leave us with more than ten thousand terms. The resulting vectors were quite sparse, which in many cases has been used to speed up the implementation. On the average there were 19.6 terms per document vector of dimension 1000.

9.8 Clustering bibliographic data.

This section describes the evaluation of the neural networks and reports the retrieval effectiveness of the Fritzke network on two standard test collections.

Relative evaluation[18]

The main difference between the artificial data discussed in section 9.5 and the bibliographic data from our database is that the bibliographic data have a higher number of dimensions, and do not form such elegant clusters. Besides that, objective relevance judgements for the collection on education were not available. So, clearly there are two important issues to be looked at: scale and evaluation.

First comes the issue of whether the networks can handle the increase in scale, and the fuzziness, of the real-world data set. Each document was represented by a thousand-dimensional term vector, compared to one hundred for the artificial data. Contrasted to the artificial data, which have a well defined notion of cluster membership, the bibliographic data are very ill-defined and noisy. The effect of the number of documents processed was looked at by looking at two settings for the collection size. The Fritzke network seemed to handle the move from an artificial to a more realistic scale well, despite the fact that long training times were needed. Gridnet was not tested in this phase, because it was felt that it did not have any advantages over the other two types of networks. The Kohonen map did not handle the scale well. On many training trials the network collapsed, i.e. a small part of the map came to represent all the patterns, and the rest did not become active. It is suspected that this difference of performance is a symptom of the fact that the Fritzke network is always more or less organised, while the Kohonen network starts out disorganised. The rest of this chapter is devoted to the discussion of the results with the Fritzke network.

The results of the clustering process for 100 documents were first tested manually for many different parameter settings. Clustering was found to be stable in a wide region of the parameter space. This does not mean that the actual clusters (contents of the lists associated with each node) were the same for each run. By visual inspection of the global organisation

[18] The experiments in this section have been performed before we had access to standardized test collections. The relative evaluation is somewhat obsolete in comparison to the evaluation on a standard test collection. I have included them to reflect a feel of the process of experimenting that has been followed.

and the comparison of the results with a hand-made clustering of the first 100 documents, the results seem to be reasonable. It is a hard problem to evaluate how reasonable the resulting clusters exactly are, when no good relevance judgements are available. For this purpose, a comparison was made to a well known best match method of retrieval, VSM with the cosine correlation measure [Salton, 1989], using the same representation as previously described, i.e. 1000 binary represented terms. It is reasonable to assume that at least a number of the documents retrieved by both methods should match. It should be noted that this comparison is only a preliminary one, as it has a number of shortcomings. Different retrieval methods can perform at the same general level of effectiveness, while still retrieving completely different sets of documents [Harman, 1993]. The numbers below should therefore be interpreted with some caution. (see also footnote 18.)

The cluster based retrieval in the network produces a limited set of documents, which was initially not sorted any further. The only ranking is that the list of documents retrieved from the BMU is ranked higher than the neighbours list of documents. Including the neighbours list increases recall at the cost of precision. To make the two methods comparable, the retrieved set of VSM had to be restricted somehow. For this purpose VSM was thresholded at a fixed number of documents. For this the average number of documents retrieved by the network was chosen. For collection size 100, this was 2 for BMU-retrieval and 7 for BMU+neighbours-retrieval. For collection size 1000, the average number of documents retrieved by the network with the BMU-retrieval strategy was 15. A perfect recall for the network was defined to be the retrieval of all documents that were retrieved by VSM. 100% precision was when the network did not retrieve any documents that VSM did not retrieve.

$$recall_{network} = \frac{|retrieved_{network} \cap retrieved_{VSM}|}{|retrieved_{VSM}|} \qquad \text{(EQ 36)}$$

$$precision_{network} = \frac{|retrieved_{network} \cap retrieved_{VSM}|}{|retrieved_{network}|} \qquad \text{(EQ 37)}$$

By this definition of relative evaluation the performance of the network is always imperfect.

Collection size	Retrieval strategy	% Recall	% Precision	Threshold
100	bmu	73	86	top 2
	bmu+neigh.	86	39	top 2
100	bmu	26	97	top 7
	bmu+neigh.	45	53	top 7
1000	bmu	38	41	top 15
	bmu+neigh.	52	14	top 15
1000	CUTOFF	62	100	0.65 cosine

TABLE 9.2: RECALL AND PRECISION OF FRITZKE NETWORK AS COMPARED TO THE VECTOR SPACE MODEL (AVERAGE OVER 10 QUERIES)

First, the two methods were compared on a collection of 100 documents. The Fritzke network was grown to 100 nodes and then pruned, resulting in slightly more than 50 nodes on the average. As one can see in Table 9.2 above, the network retrieved the documents that the VSM had ranked top-2, in 73% of the cases. When more correspondence (top-7) is required the recall drops considerably, but precision is maintained at high levels.

Increasing the size of the document collection to 1000 documents, results in a large drop in recall and precision. This was done with approximately the same number of units in the network as for size 100. The decrease is serious, but it does not look like a total collapse. Part of it is due to the fact that the average number of retrieved documents is quite large, because each unit of the network holds much more documents. On such a large retrieved set perfect correspondence with VSM becomes very unlikely. Performance can be improved after clustering by choosing a different retrieval strategy and thus de facto artificially reducing the retrieved set of the network and of VSM (CUT-OFF-strategy in the table above). This was done by ranking the documents *within a selected cluster* by the cosine coefficient, and setting an absolute threshold of 0.65 for similarity. In this case, if a node contains a lot of documents, but they are all very different from the query, they are not retrieved. The improvement of performance in this last case indicates that the clusters do contain the right documents, but also many irrelevant ones. This should be expected to get better when a finer cluster structure (more nodes) is trained. In the case of the Fritzke network this simply means growing the network for a longer number of time steps.

Absolute evaluation

To asses the quality of the clusters produced by Fritzke's growing cell network objectively, we have tested them on two standard test collections. Intuitions about which variables control the behaviour of the network in an IR setting, suggested by the set of experiments on our initial data, are investigated in more detail here.

The test collections provide a set of documents, a set of queries, and for each query the numbers of the documents judged to be relevant by humans. This allows an automatic comparison of the output of the network with the "right" answers, thus avoiding the problems with relative comparison. The Keen and Cranfield collections have been used extensively for the comparison of clustered retrieval in the literature. [Griffiths *et al.*, 1986] tested hierarchic agglomerative clustering algorithms, and [MacLeod and Robertson, 1991] their neural clustering algorithm on the same two test collections. Besides these touchstones from the literature a simple vector space model was run on the test collections with two levels of retrieval volume. If the cluster hypothesis is to be useful, the best match search of the vector space model should be surpassed in effectiveness by cluster based retrieval. This was not the case in our experiments. The characteristics of the collections are summarised in Tables 9.3 and 9.4 below.

TABLE 9.3: CHARACTERISTICS OF THE CRANFIELD COLLECTION.

Documents	1400
Terms	2557
Terms/Document	28.7
Queries	225
Terms/Query	8.0
Relevant docs/Query	7.2
Description	Documents characterised by lists of manually assigned index terms. The subject is aerodynamics.
Notes	Of all available test collections this one is known to be best susceptible to clustering.

TABLE 9.4: CHARACTERISTICS OF THE KEEN COLLECTION.

Documents	800
Terms	1432
Terms/Document	9.8
Queries	63
Terms/Query	10.3
Relevant docs/Query	14.9
Description	Document titles, augmented by manually assigned index terms. The subject is librarianship and information science.
Notes	Known to be less susceptible to clustering.

On these two test collections we conducted a number of experiments. The variables were network size and training time. The only network type used was Fritzke's. The methods used for clustering and retrieval are similar to the methods used on the artificial data set in section 9.5. The network was trained on a randomised presentation of the patterns, the clusters were compiled, and the queries from the collection were presented. A notable difference was the fact that in the test collections the queries are not part of the training material, and that the queries are typically much shorter than the documents used for training. The Euclidean distance measure, which is the standard choice for the Fritzke network - was unable to find the correct matches, because it is not normalised with respect to vector length. Therefore, the Euclidean distance was used during the training phase, but the queries were matched to the weight vectors using cosine correlation. We have only used the BMU-retrieval strategy, although an estimate will be presented of how well other strategies would have performed. For each query in the collection the retrieved set was compared to the relevance judgements, i.e. to the "right" answers. From this, average recall and precision figures, as well as the combined effectiveness score, using the *E*-measure, were computed.

The *E*-measure was introduced by [Van Rijsbergen, 1979], in order to be able to measure the effectiveness of an IR system by one number. It combines recall (*r*) and precision (*p*).

$$E = 1 - \frac{(1+\beta^2)p \cdot r}{\beta^2 p + r} \qquad \text{(EQ 38)}$$

The parameter β reflects the relative importance attached to recall and precision. A value of 0.5 (or 2.0) for β corresponds to attaching twice (or half) as much importance to precision as to recall. With a value of 1.0 for β, the two factors are weighted equally. *E* has a range between 1 and 0, and lower values of the *E*- measure indicate higher performance. So 1 is the worst, and 0 is perfect.

Results on Cranfield and Keen collections

This extended series of experiments was designed to answer four questions about the behaviour of the networks.

- How well does the Fritzke network do in relation to other well-known methods?
- How long must the network be trained in order to yield good clusters?
- What is the effect of the size of the network on retrieval effectiveness?
- Is the topological organisation of the network a valuable tool for browsing?

These matters are to some degree interrelated. While searching for answers, a fifth question arose.

- How much variation is there between the results of two different runs of the network, initialised with the same parameters?

In this section we will try to give answers to all of these questions. The discussion uses the material depicted in the series of figures below. The experimental data from which these figures have been made can be found in [Zavrel, 1995]. An important point for interpretation of the figures is the following. For each collection there are three graphs, one for each value of parameter β[19]. Each figure depicts the effectiveness of three differently sized networks, measured by the *E*-value, as a function of training time. Remember that a lower *E*-value means better performance. The points on the graph *do not* represent the performance of the network as sampled during training. Each point is a different and fully trained network. The Fritzke network grows from a small initial state to its final size in a fixed schedule.

The effectiveness of the networks is measured at three settings of the β parameter. The dotted lines indicate the level of effectiveness of methods of comparison, tested on the same collection. networks were trained for 1,400, 2,800, 5,600, 10,000, 20,000, 40,000 and 100,000 time steps. For the networks with size 250 on the Cranfield collection (Figure 9.11), all experiments were repeated three times, with different random weight initialisations. On the smaller Keen collection the networks were trained for 800, 1,600, 3,200, 10,000, 20,000, 40,000 and 100,000 time steps.

[19] β=0.5 measures precision oriented search, β=2.0 measures recall oriented search, and β=1.0 measures a n evenly weighted combination of the two. The graphs with β=1.0 give the best indication of the overall performance of the network.

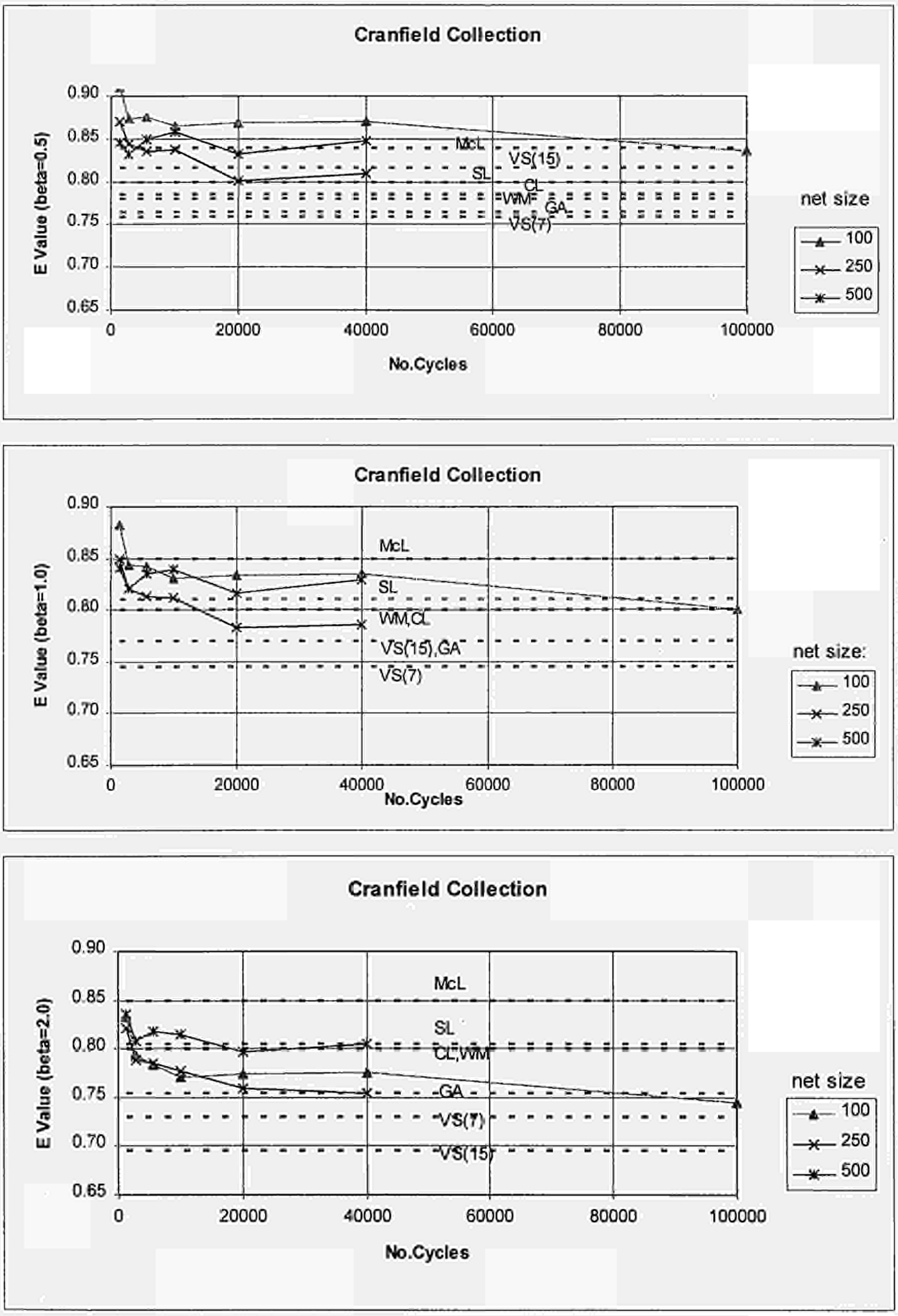


FIGURE 9.11: NETWORK PERFORMANCE (FRITZKE) ON THE CRANFIELD COLLECTION.

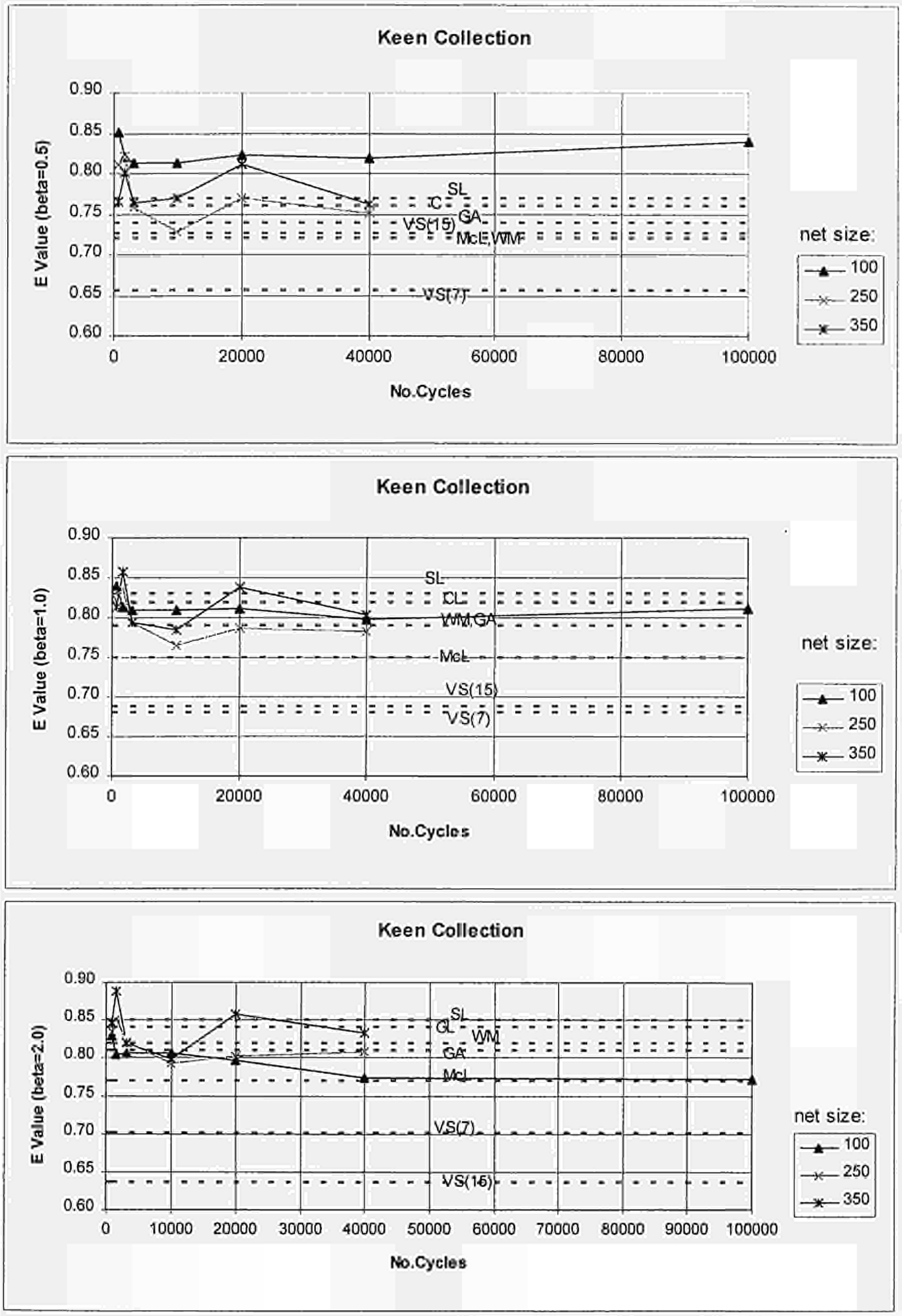


FIGURE 9.12: NETWORK PERFORMANCE (FRITZKE) ON THE KEEN COLLECTION.

The effect of choosing a larger number of time steps to train a network of the same size is that the interval between insertions of new cells (λ, in Appendix A) is increased. Besides this, of course, longer training time means that each pattern is presented more often. So, when a line in the graph shows worse performance after more training steps, this does not mean that the same network became worse after longer training, but that a differently initialised network of the same size turned out worse after its own training period.

The network in relation to other methods

The effectiveness of the networks in comparison to the methods in [Griffiths *et al.*, 1986, Table 2.], and in [MacLeod and Robertson, 1991] is shown in Figures 9.11 and 9.12. Effectiveness varies with time and network size, which shall be discussed below, but in general it is within the same range as the other methods. The methods of comparison are the following. VS or VSM is the vector space model (with the cut-off in parentheses), McL is the MacLeod algorithm, SL is single linkage, CL is complete linkage, GA is group average, and WM is Ward's method. These latter four methods are computationally expensive hierarchic clustering algorithms. They are drawn as dotted lines in the graphs.

		E-value		
Collection	Algorithm	β=0.5	β=1.0	β=2.0
Cranfield	VSM (vector1)	**0.76**	**0.75**	**0.73**
	Fritzke (cranet5b)	**0.76**	**0.75**	0.74
	McL	0.84	0.85	0.85
	SL	0.80	0.81	0.81
	CL	0.79	0.80	0.80
	GA	0.77	0.77	0.76
	WM	0.78	0.80	0.80
Keen	VSM (vector1)	**0.66**	**0.68**	**0.70**
	Fritzke (keenet04)	0.73	0.77	0.79
	McL	0.72	0.75	0.77
	SL	0.77	0.83	0.85
	CL	0.76	0.82	0.84
	GA	0.74	0.79	0.81
	WM	0.72	0.79	0.82

TABLE 9.5: COMPARISON OF THE BEST PERFORMANCE OF THE NETWORK TO OTHER METHODS.

A surprising finding is that VSM outperforms all the cluster algorithms. This shows that the cluster hypothesis is really just a hypothesis. To make a simple comparison possible, The table above shows the effectiveness of the best networks trained next to the numbers for the other methods. From this table we can conclude that under the best possible circumstances the Fritzke network outperforms all other clustering methods for the Cranfield collection. Only VSM surpasses it here. On the Keen collection the performance of the Fritzke network is again superior to the hierarchical agglomerative clustering algorithms. Here it performs worse than VSM and McL. A look at Figure 9.11 for the Cranfield collection reveals, however, that for those experiments which have been replicated three times (those with network size 250) the average performance at its best point is not as good as the best performance from the table. In fact, as will be discussed in below, network performance shows quite a lot of variation from random influences. So the best network might very well be just a lucky hit.

Quality of clusters and training time

The training of the network is a slow tuning of the weight vectors by exposure to the data. It is clear that the amount of exposure is important for the quality of the tuning process. Intuitively, one would expect that more training simply is better. However. the figures reveal a more complicated story. For both collections the smallest networks, those with size 100, are the easiest to explain. Going from one pass over the data set (1400 time steps for the Cranfield collection) to two passes results in a steep increase of performance. The increase of performance per extra pass then becomes smaller and smaller. For the Cranfield collection it seems that it still has a chance for improvement by going beyond 100,000 training steps. For the Keen collection, which is smaller, the performance increase levels off, and it seems like there simply is no more room for improvement in a network of size 100.

The larger networks differ from this simple story. Although the general tendency is to improve with more training time, this does not always hold. E.g. in the graph it looks like 20,000 training steps is particularly bad choice for the Keen Collection. We have not been able to determine what other process than chance might have caused this behaviour. It might be that larger networks, which have more unit insertion steps suffer more from the randomness of the pattern presentation. Or maybe a larger network simply needs much more pattern presentations than a small one.

Despite these fluctuations and the other random effects it seems as though a relatively small number of passes through the collection suffice to bring the network into a desirable region of performance. Hereafter the change is not so significant anymore, uncertain and computationally expensive[20].

Network size and retrieval effectiveness

The effect of network size on effectiveness depends on the choice between recall and precision. A large network forms many small clusters and does not return a large volume of

[20] A simulation of a 100 unit network for 100,000 timesteps took more than a week of PC-time.

documents for a query. Therefore it is unlikely to score very high on recall. However, the clusters are better suited for a tight fit to the data, so precision is favoured. This can be seen in the comparison between size 100 and size 250 networks. Size 250 is significantly better in the precision oriented (β=0.5) measurement on both collections. Although in the recall oriented measurement size 100 does very well, this gives size 250 enough of an edge over 100 that it does better in the averaged measurement. A strange phenomenon is displayed by the largest networks. Although they do take part in the initial drop of the *E*-value in the graph, they consistently go up again with more training. This again points in the direction that large networks simply need *much more* training than small ones. It should be hoped that their performance will eventually go up at 100,000 steps, but this was not yet confirmed, for lack of computation time. If this will not happen, they can be considered a waste of resources.

Utility of the topological organisation

The effectiveness figures discussed above were all derived from a single cluster, i.e. BMU-cluster, retrieval method. If the browsing functionality of topologically organised networks is to be taken seriously, the neighbourhood relationship between units in the network should reflect similarity in the document collection. Browsing the neighbours of the BMU should therefore improve effectiveness. We have not tested this on a large scale, but Figure 9.13 gives an indication of the effect of browsing. The figure displays recall and precision values after browsing *n* neighbours, averaged over many queries. It is assumed that the typical browsing action will go through the contents of the neighbouring clusters in order of their distance to the BMU of the query. This ordering can be obtained through a ranking by the distance between the units' weight vectors. As can be seen in the figure, recall does considerably improve by browsing, but one should wonder whether the drop in precision is not too high a price for this. We expect that further within-cluster ranking should make the drop in precision less dramatic. At any rate, the neighbourhood relation in the network is to some degree relevant for IR purposes.

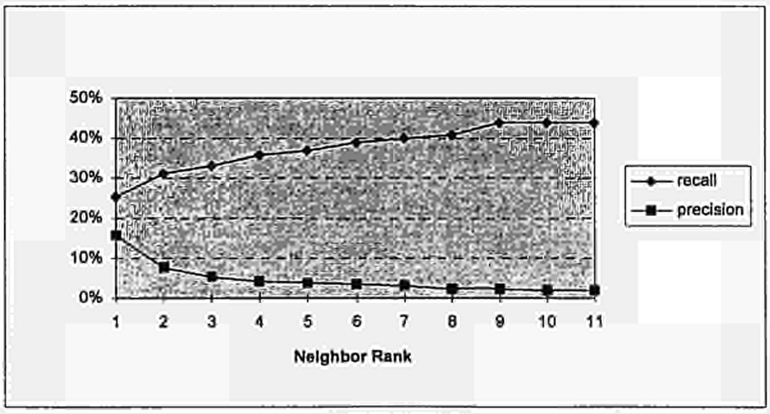


FIGURE 9.13: THE EFFECT OF BROWSING THE NEIGHBOURS ON RETRIEVAL EFFECTIVENESS.

Variation and reliability

During the experiments it became clear that starting out with the same settings for size and training time does not necessarily result in the same trained network. Not only is the process non-deterministic, but its results turned out to be very variable. This is caused by two random factors in the network training. In the first place, the weights of the network are initialised with random values. In the second place, the patterns are presented in a random order. The extent of the variability can be seen in Figure 9.14 below, where the 95% confidence intervals are plotted, as estimated from three replications of each experiment[21]. It seems as though the variance is large enough to cause the "bumps" in the series of graphs above. Nevertheless, the tendency of the clusters to become better with more training time remains. A factor that might have amplified the random effects in the network is the fact that in our simulations the pruning of the network was only done after the whole training phase. If useless parts of the network are not pruned out in time the rest of the network may be hampered in its development.

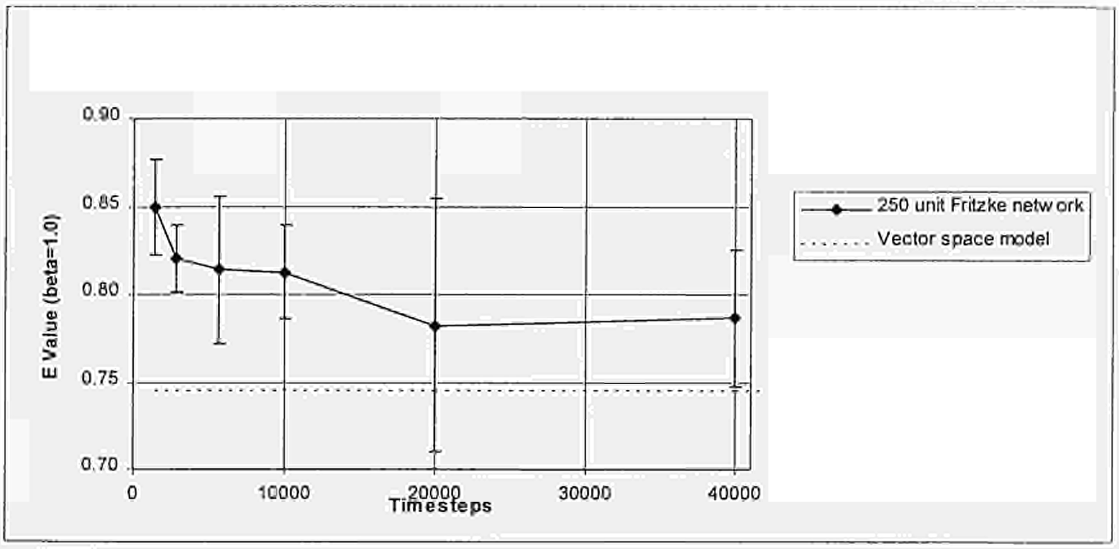


FIGURE 9.14: RELIABILITY OF CONVERGENCE OF A 250 UNIT FRITZKE NETWORK ON THE CRANFIELD COLLECTION.

[21] The 95% confidence intervals have been computed assuming a normal distribution and fitting it to the outcome of the three experiments for each data point. It should be noted that this estimate gives much larger variance than an estimate for a larger (>>3) sample size with the same mean.

9.9 Discussion

The aim of this chapter was to ascertain whether neural networks for clustering and browsing are a promising research subject. Are neural networks still promising in this domain? The experimental results reported in the section 9.8 suggest that the clusters produced by the Fritzke network are on par with the clusters produced by the methods of comparison. However, they are not consistently better, due to the unreliability of the training phase. Nonetheless, a number of interesting points have emerged. Many of the early problems of ANN's for bibliographic clustering, most notably those of scalability and cluster separability, have been shown solvable. The main problem seems to be the unreliability of convergence. After analysis and experimental evaluation neural networks lose some of their intuitive appeal. But, this is the only way to go. , As discussed in section, further research should focus on the issue of reliability of convergence and the issue of utility of the topological organisation for browsing purposes.

10 Filter Prototype

-- Marco-René Spruit

Introduction

The approach taken in FILTER consists of matching incoming full-text data, such as news and abstracts, to a neural network representing a specific user interest. Only data correlating with this interest is returned to the user. This is known as selective dissemination of information (SDI) or the filter principle.

The amount of information which is stored in libraries is growing exponentially. This exponential growth makes it virtually impossible to maintain a manually structured database for all incoming data. Also, information storage sources are moving from analogue form towards digital form. These shifts make the automatic signalling and distribution of incoming information by computer services respectively increasingly important and more generally applicable. This seems an obvious task for libraries, being genuine information archivers.

The present chapter is divided into three parts. First, a basic theoretical background is provided to explain the context of the project. Next, the prototype is described by reviewing the implementation of some generally important application properties and by an example of an imaginary session. Finally, it is explained how the prototype has been evaluated and what conclusions can be drawn, based on this evaluation.

This entire chapter is also available within the FILTER prototype itself as part of the on-line documentation.

10.1 Background

Information retrieval

Information retrieval is the matching of a query against a large number of documents. Two types of application environments can be distinguished in this field:

- A relatively static database environment which is investigated with dynamic queries. This is known as free-text search or document retrieval.
- A dynamic database environment which needs to be filtered with respect to relatively static queries. This is known as the filtering problem, current awareness or selective dissemination of information.

In a static database environment, the user formulates a query which is being matched against the documents in the database and the proper texts are returned to the user within seconds. A query in this context consists of keywords with optional wild cards. Its internal relations can be controlled by logical and statistical operators[22]. The data corresponding to a query can be retrieved very quickly, because the data collection has been pre-processed by generating an index over the database. An index contains all unique strings in the data collection, together with their positions in each document. Therefore, the index can be searched instead of the unordered database, which is virtually infinitely faster.

In a dynamic database environment, the user formulates a query or subscribes to an existing one, which corresponds to his or her personal interest. All incoming data is matched against these profiles and the proper texts are distributed to the user periodically. Although a query in this context is in principle syntactically identical to a query in a static database environment, this query's semantics is essentially different. In this context, a query's connotation resembles a user profile or interest description, which has a more enduring character. The incoming data must be indexed first before it can be matched and distributed according to the profiles. Once the index has been generated and stored in the database, together with the original data, the database environment becomes static for this passed period of time.

The user plays an active part in the retrieval process in a static database environment. In a dynamic database environment, the user formulates only once what he or she wants to be retrieved, for as long as the given profile corresponds with his or her interest.

There are some serious drawbacks in index-based information retrieval:

- For an average user, it often turns out be quite difficult to formulate a query which accurately corresponds to his or her intentions.
- Only documents which contain the query-keywords can be retrieved[23]. It has no ability to generalise over a query and cannot make incomplete matches well[24].

[22] Common logical operators are the conjunction (AND), the disjunction (OR) and the negation (NOT). An example of a statistical operator is the quorum (n of $\{k_1, k_2, ...\}$). This means that n keywords of the keyword-set must be in the document.

[23] To optimise retrieval, thesauri, or synonym-vocabularies, can be included in contemporary IR applications. However, this is not a real solution. The danger of retrieval-overkill increases significantly.

- No real context can be incorporated[25].

It would mean a large step for the field of IR to have a method which could incorporate these shortcomings without slowing down.

Artificial neural networks

Artificial neural networks are mathematical pattern recognition models which, although a variety of neural network architectures have been developed, all exhibit some interesting properties, important in an information retrieval context:

- Distributed data representation, i.e. x objects[26] are represented by y neurones. The ability to generalise over the data increases as the ratio between the number of objects, presented to the feature map, and the number of available neurones increases[27].

- Robust behaviour, i.e. the ability to process incomplete or incorrect information. This is a consequence of the distributed data representation and the ability to generalise over the data.

- Language independence, i.e. not the data itself is used during the process, but an internal vector representation, which is a series of numbers, representing a co-ordinate in the vector space.

From the large variety of neural network architectures, the Kohonen feature map has been implemented in the FILTER prototype.

[24] In contemporary IR applications a fuzzy search can be performed though. But since this fuzzy algorithm generates a number of keyword-permutations autonomously of the context, it can easily result in a retrieval-overkill.

[25] There do exist some context sensitivity-imitation techniques. An example is the binary proximity operator (*A* within *n* words of *B*), which searches for two keywords (*A* and *B*) within a range of *n* words. However, this is an incorporation of context in the statistical sense and not in the semantical sense.

[26] An object can, for example, be a natural language character or word.

[27] Architectures that do not have this property are not considered neural, but statistic table models.

Kohonen feature map

The Kohonen feature map is known to be an abstraction of the biological topology preserving maps found in the human visual system. It can be thought of as a two-dimensional grid. Each node in the grid contains a neurone, i.e. a set of input fibres or sensors or a vector representing a co-ordinate in the vector space. The data, which is trained to the feature map, is translated into vectors before it is presented. Therefore, the two-dimensional feature map can capture that portion of the multi-dimensional vector space which corresponds to the internal vector representation of the data.

The Kohonen formalism is a competitive learning algorithm. The two-dimensional feature map is a rectangular or hexagonal structure of neurones, which all have the same number of weights. The activation of a neurone, resulting from an input activation, is interpreted as a measure of correlation. The neurone, best representing the input activation, can therefore be determined by finding the neurone with the highest activity. In other words, the neurone, best representing the input vector, can be determined by finding the map vector with the minimum mathematical distance[28] with respect to the input vector. This neurone is called the best matching unit[29].

Once this neurone has been found, neurones within a certain region are adapted to some extent, depending on their distance from the best matching unit. Therefore, this region will recognise the current input better in the future. In time, the adaptation value and the region adaptation size also decrease to guarantee convergence. Because neighbouring neurones are updated with respect to an input vector's best matching unit each training cycle, a topological map emerges, holding related data elements in neighbouring regions[30].

To summarise, the feature map has some additional interesting properties, besides the general artificial neural network properties:

- Self-organisation on frequency and context, i.e. the frequencies of input patterns and overlaps between parts of these patterns, i.e. the patterns' context, are equally important.

[28] The most commonly used mathematical distance in vector space models is the Euclidean distance. This distance has also been used in this prototype.

[29] The common abbreviation for the Best Matching Unit is BMU.

Therefore, this automatic feature extraction out of unstructured data results in a map of conditional probabilities.

- Unsupervised training, i.e. the representation process of the training data in the feature map is fully automated. Therefore, one does not need to have any knowledge of the system architecture to be able to use such a system, if the system parameters are pre-configured.
- Topology preservation, i.e. if two object vectors are close to each other in the vector space, they will also be close to each other in the feature map after the training process. This results in a natural clustering of data features.
- Also, the Kohonen formalism is computationally efficient with respect to other neural architectures and it is relatively easy to implement.

Alternatives

Adaptive Resonance Theory (ART) [Carpenter *et al.*, 1991c] also encapsulates self-organisation and unsupervised training in a more neurobiologically founded manner. By integrating two subsystems, of which the higher-level subsystem supervises the lower-level subsystem, a stable and dynamic neural environment can be created. However, the working becomes quite complex, due to the many parameters involved. Also, the algorithm is computationally expensive.

The Simple Recurrent Network (SRN) [Elman, 1990] uses a recurrent network, where the hidden layer units are fed back into the input layer. Training such a network can also be considered unsupervised. By using recurrent input fibres, the model implements a higher order Markov chain[31]. Therefore, the network will contain a too specific representation of the data after the training process. Another known problem is the long training time, making it computationally expensive.

Other common neural architectures lack self-organisation and unsupervised training. These properties are important in the filtering problem, since it is not known in advance what ought

[31] A Markov chain is a mathematical model for event prediction. It was developed by A.A. Markov. The method is used to predict the possibility of an occurrence, given a history of occurrences. In general, in a high order Markov chain, an event can be a complex function. In the case of natural language data, the order of the Markov chain represents the number of preceding characters which are used to predict the next character.

to be trained. For this reason the Kohonen feature map was implemented for the FILTER prototype.

Neural filter

The neural filter algorithm implements a mechanism in which a query or user interest or profile, stated in natural language, is taught to a self-organising neural network, which derives an internal representation of the text. This representation is then matched against a continuous stream of incoming, unstructured data. The general set-up can be seen in Figure 10.1 below. Optionally, multiple queries can be matched simultaneously.

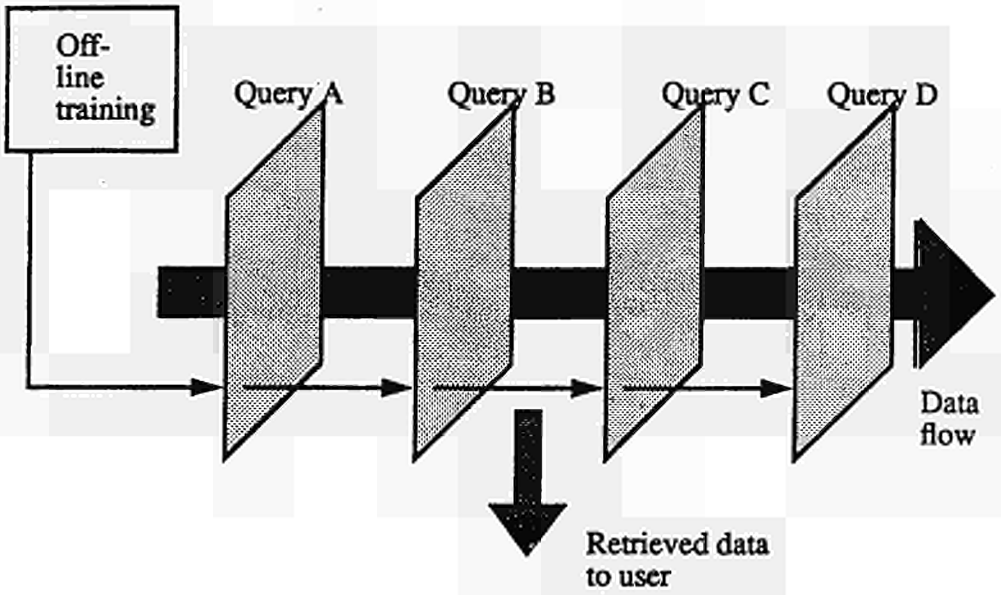


FIGURE 10.1: PRINCIPLE OF THE NEURAL FILTER (REPRINTED FROM [SCHOLTES 1993]).

Several algorithmic variants are possible, depending on the choice of objects which are presented to the neural network. In this project, characters have been used as basic objects to automatically incorporate context and maintain a more direct language independence[32]. This

[32] If words would have been used as objects, a dictionary would have been necessary. Such a dictionary could have been generated in advance though, by a statistical frequency algorithm. However, in the FILTER prototype language-dependent noise-

variant is known as the *n*-gram analysis method (see Figure 10.2) A *n*-gram is a *n*-length sequence of characters[33]. The *n*-gram analysis method can be interpreted as a window of size *n*, which is being shifted over the text. It is implemented in the Kohonen input sensors by assigning several sensors to each object within the window and concatenating all the window sensors to one big input vector. By shifting this window over the training text, only frequent *n*-grams form clusters on the feature map, infrequent patterns are overruled.

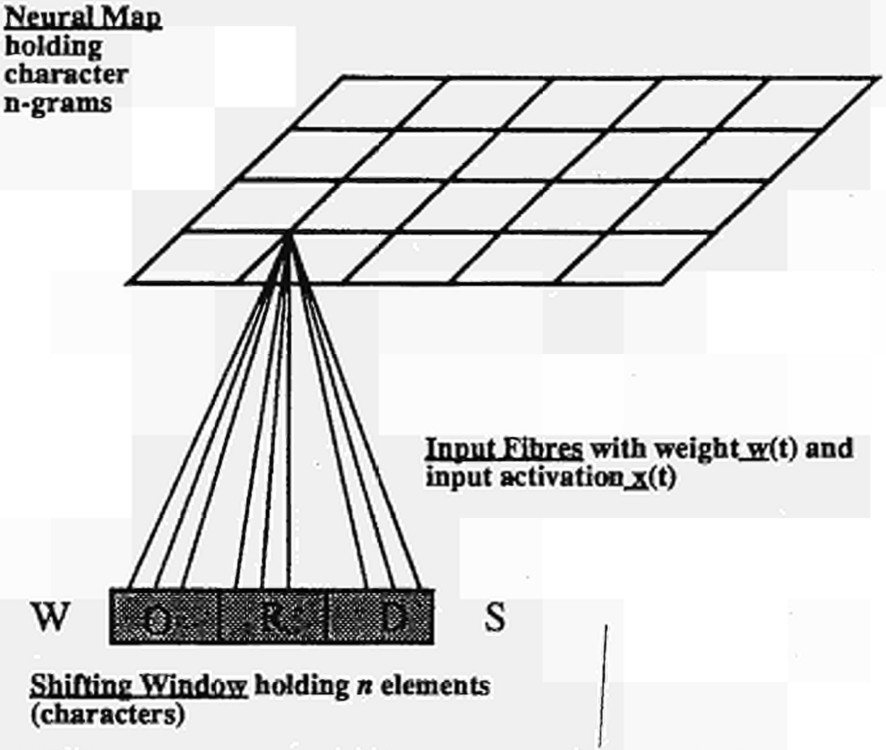

Neural Map holding character n-grams
Input Fibres with weight w(t) and input activation x(t)
W
O
R
D
S
Shifting Window holding *n* elements (characters)

FIGURE 10.2: N-GRAM ANALYSIS METHOD (REPRINTED FROM [SCHOLTES 1993]).

After training, the input values of texts, mediated through the shifting window, which correspond to the query representation in the feature map, will yield low normalised cumulative errors and a high number of normalised cumulative perfect hits. This means that there is a certain degree of resemblance, or correlation, between these two texts. Therefore, if the feature map is used this way, it can be incorporated as a filtering device in an environment with relatively static queries and a dynamic information flow. This approach can also resolve some of the drawbacks of index-based retrieval.

words and noise-word endings can be eliminated to optimize performance. But, these noise-lists could also be generated in advance by a statistical frequency algorithm.

[33] For example, the set of possible trigrams, i.e. n=3, with the word TRIGRAM is : ??T, ?TR, TRI, RIG, IGR, GRA, RAM, AM?, M??}, where ?'s indicate variable context characters.

The schematic version of the algorithm is given in Table 10.1 below:

- **Initialise objects**

- **Initialise feature map**

- **Initialise input sensor**

- **Initialise text part statistics**

- **Teach query to feature map**
    - Filter data in chunks
    - Eliminate non-alphabetic characters and separate all words with a space character
    - Convert lower case characters to upper case
    - Optionally eliminate non-relevant words, non-relevant word endings and space characters
    - Shift window over filtered data to determine n-gram patterns
    - Convert patterns to vectors and copy in the input sensor
    - Present input sensor to feature map
    - Determine BMU
    - Determine current map region size to be updated
    - Determine current learn rate
    - Adjust the region of the BMU

- **Extract text parts from data flow**
    - Filter data in chunks
    - Eliminate non-alphabetic characters and separate all words with a space character
    - Convert lower case characters to upper case
    - Optionally eliminate non-relevant words, non-relevant word endings and space characters
    - Shift window over filtered data to determine n-gram patterns
    - Convert patterns to vectors and copy in the input sensor
    - Present input sensor to feature map
    - Determine the error of the BMU
    - Update text part statistics
    - Determine correlation between query and text part, based on the text part statistics

TABLE 10.1: SCHEMATIC VERSION OF THE NEURAL FILTER ALGORITHM.

10.2 Prototype

In this section we discuss the application properties on which we focused during prototype development: flexibility, performance, visualisations of the feature map and the accessibility issue.

An overview of the prototype in the form of an imaginary session will be given as well.

Properties

During the prototype development, four application properties were considered essential to create a properly functioning neural filtering environment:

- Flexibility, i.e. the ability to adjust and execute any valid event at any time to render interactive research.
- Performance, i.e. the speed at which accurate retrieval can be achieved.
- Visualisation, i.e. the clarification of the processes and the data by viewing these from different perspectives.
- Accessibility, i.e. the storage and retrieval of all input and output to enable reconstructions and variations.

Flexibility

Flexibility is of importance to interactive research. One has to be able to efficiently experiment with the process parameters.

An event-driven, multitasking environment is needed to provide maximum user-interactivity. This implies an Object Oriented Programming (OOP) concept.

In FILTER, every event can be fine-tuned, or even redefined, at *any* point in *any* process by a set of parameters and preferences.

Performance

Performance stresses the importance of execution speed in this type of applications. The importance of the accuracy of the retrieval is merely implicitly accentuated here, because this has been considered an obvious goal to achieve. However, if accurate retrieval cannot be achieved at a high speed, the system will simply not keep up with the incoming data flow in a real-time filtering situation. Then, the system would still be useless, regardless of its retrieval accuracy.

The filtering process consists basically out of four continuously repeated events:

- Read the incoming data.
- Convert the data to patterns.
- Convert each pattern to its vector representation.
- *Search* the nearest neighbour in the feature map for each input vector.

The most time-consuming event in a single processor[34] environment is the nearest-neighbour search, because for each input vector the whole feature map must be searched. Therefore, two possible optimisations have been investigated to speed up this event.

Dynamic k-d tree

A k-d tree is a tree structure for storing and retrieving k-dimensional data points. Although the k-d tree is usually being built during training, it should also be possible to convert the feature map into this structure after training by dividing the feature map recursively into two equal neurone collections along the axis of greatest range. The data points are stored in the leaf nodes. By using this search technique, the search time decreases exponentially[35], if the tree is in balance. See also Figure 10.3.

[34] The conceptually most obvious minimisation of execution time, i.e. the implementation in a multiprocessor environment, has been ignored in this report for practical reasons.

[35] The full-search algorithm has a complexity O(N), where N is the number of neurones in the feature map. The tree-search algorithm has a complexity O(log N).

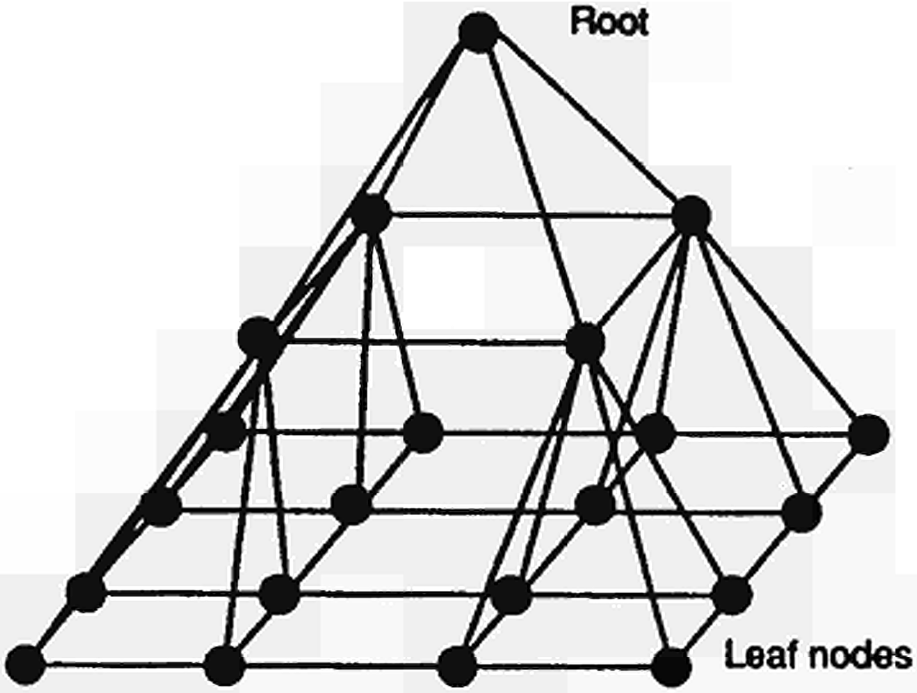


FIGURE 10.3: FEATURE MAP ELEMENTS AS LEAF NODES IN A TREE STRUCTURE (REPRINTED FROM [KOIKKALAINEN 1990]).

In the implementation, each internal node contained an average vector of the data co-ordinates of its two daughters. Unfortunately though, it turned out that these average vectors levelled too much after a few tree levels. Relatively often, this led to inaccurate retrieval of a pattern's best matching unit.

Although the k-d tree structure could be an efficient representation of the feature map, the nearest-neighbour search would still dominate the filtering process as the most time-consuming event. The distance between each input vector and some map vectors must still be calculated. Therefore, another approach called the *Table map* was tried.

Table map

Hashing is a well known statistical addressing technique which retrieves the output by calculating a function of the input. In this case, this means that the distance of the best matching unit must be returned, based on the input patterns. To accomplish this, all possible patterns must be generated, each pattern must be matched against the feature map and each distance must be stored at the position in the hashing object which represents its pattern. This can take some time, but it has to be done only once for a trained feature map. This way, the actual filtering process events can be replaced by:

- Converting each pattern to its hashing address.
- *Get* the distance, contained in that address.

However, depending on the context size, there can be very many possible patterns[36]. To enable storage capacity for up to a hundred million distances, a two-dimensional hash table was implemented. In the prototype, this is called a table map, analogous to the feature map and the vector map. In practice though, a table map with a hundred million entries is not efficient anymore. It takes a lot of memory and preparation time[37]. Therefore, if the table map is to be used, the context size should be kept low. In the evaluation section, it will be investigated whether a low context size is possible or not, in relation to the retrieval accuracy.

Visualisation

In dealing with visualisation, the importance of viewing the data from different perspectives is stressed, in order to help us clarify what exactly is happening with the application objects.

Two types of data visualisation have been implemented:

- Textual visualisation, i.e. a print of the contents of an object in ASCII-format. This can be useful as a low-level clarification source.

[36] The number of possible patterns is O^c, where O is the number of characters in the language and c is the context size.

[37] The amount of memory needed for each table, containing 10000 Euclidean distances, is 80 Kb. This means that if the number of entries is $27^3 = 19683$, the table map will take 160 Kb of memory. (Generation will take about 8 minutes, depending on the feature map size.) If the number of entries is $27^4 = 531441$, the table map will already consume 4.3 Mb of available memory. Preparation time here includes generation, saving and loading time.

- Graphical visualisation, i.e. a print of relations within an object in a unrestricted format. This is very useful as a higher-level clarification source.

Both textual and graphical visualisation can reflect two different perspectives on an object:

- Static perspective, i.e. a print is a snapshot of an object.

- Dynamic perspective, i.e. the print is an anchored view onto an object to follow the process.

Textual visualisation

Among the static textual visualisations are the feature map which can be printed to view the neural weights and the vector map which be printed to clarify the internal vector representation of the data. Also, the contents of the error- and activity recording objects can be printed to enable customised visualisation with an external spreadsheet application.

Dynamic textual visualisation has been applied to the internal data flow of both the query and the passing data to examine how exactly the input is transformed, before it is presented to the feature map.

Below is a fragment of the textual visualisation of the contents of a feature map. Each neurone in the feature map consists of ((Sensors per object)*(Context size)) weights. These concatenations of weights are used as vectors, representing co-ordinates in a multi-dimensional space. This example shows a cluster in the fifth and sixth dimension of several neurones, which means that this region will recognise patterns, which have a character in the middle which maps to 1.000000 (see also the vector map in Table 10.3).

```
FEATURE MAP initialised with current settings...
            X-dimension                  : 13
            Y-dimension                  : 17
            Context size                 : 3
            Sensors/Neurone              : 9
            Random spread                : 15
neurone[0,0]        : 0.224287 0.952296 0.029184 0.029922 0.687756 0.486086 0.873835 0.900448 0.889078
neurone[0,1]        : 0.127397 0.976929 0.028571 0.372500 0.536198 0.871545 0.803125 0.985772 0.996528
neurone[0,2]        : 0.303389 0.937831 0.002000 0.496275 0.509724 0.998167 0.492377 0.996237 0.997918
neurone[0,3]        : 0.377346 0.997539 0.003503 0.542813 0.995419 0.999993 0.058903 0.779591 0.805812
neurone[0,4]        : 0.697131 0.958209 0.025608 0.967122 0.999992 0.999998 0.091885 0.198423 0.993085
neurone[0,5]        : 0.558126 0.403755 0.226493 0.997551 0.999998 0.999998 0.006682 0.016472 0.949989
neurone[0,6]        : 0.974968 0.136078 0.503579 0.994403 0.999998 0.999998 0.125672 0.570563 0.768181
neurone[0,7]        : 0.697058 0.043202 0.876233 0.807937 0.999998 0.999998 0.040433 0.202237 0.363312
neurone[0,8]        : 0.343057 0.355185 0.932372 0.756739 0.999777 0.999998 0.808857 0.021074 0.368147
neurone[0,9]        : 0.096726 0.503642 0.512291 0.715875 0.999692 0.999998 0.993122 0.076557 0.223334
neurone[0,10]       : 0.597828 0.829262 0.429358 0.926244 0.999499 0.999998 0.791958 0.497056 0.106886
neurone[0,11]       : 0.502821 0.997213 0.978075 0.999931 0.999998 0.999998 0.449903 0.631282 0.184550
neurone[0,12]       : 0.685060 0.869337 0.995705 0.894068 0.902305 0.999998 0.661747 0.613795 0.036563
neurone[0,13]       : 0.798250 0.979789 0.998584 0.509939 0.576002 0.999993 0.676429 0.866105 0.002862
neurone[0,14]       : 0.922398 0.594168 0.717885 0.372101 0.523254 0.831094 0.591289 0.945196 0.592083
neurone[0,15]       : 0.988563 0.380237 0.419349 0.022776 0.403899 0.420415 0.513318 0.999918 0.999917
neurone[0,16]       : 0.901105 0.286195 0.012265 0.066698 0.544840 0.558795 0.600750 0.999992 0.999998
neurone[1,0]        : 0.242177 0.625855 0.311376 0.016595 0.297078 0.142991 0.847766 0.988551 0.100470
neurone[1,1]        : 0.084173 0.956070 0.631514 0.031282 0.485015 0.505438 0.518370 0.998363 0.836606
neurone[1,2]        : 0.295684 0.815231 0.462251 0.262446 0.609304 0.773652 0.245971 0.922429 0.826050
```

TABLE 10.2: FEATURE MAP: TEXTUAL VISUALISATION.

The vector map below defines the mapping for each data pattern into its internal vector representation. The example is from an artificial data set reflecting the order of the alphabet.

```
VECTOR MAP initialised with current settings...
            X-dimension                  : 27
            Y-dimension                  : 3
            Code spread                  : 2
A           : 0.000000 0.000000 0.000000
B           : 0.000000 0.000000 0.500000
C           : 0.000000 0.000000 1.000000
D           : 0.000000 0.500000 0.000000
E           : 0.000000 0.500000 0.500000
F           : 0.000000 0.500000 1.000000
G           : 0.000000 1.000000 0.000000
H           : 0.000000 1.000000 0.500000
I           : 0.000000 1.000000 1.000000
J           : 0.500000 0.000000 0.000000
K           : 0.500000 0.000000 0.500000
L           : 0.500000 0.000000 1.000000
M           : 0.500000 0.500000 0.000000
N           : 0.500000 0.500000 0.500000
O           : 0.500000 0.500000 1.000000
P           : 0.500000 1.000000 0.000000
Q           : 0.500000 1.000000 0.500000
R           : 0.500000 1.000000 1.000000
S           : 1.000000 0.000000 0.000000
T           : 1.000000 0.000000 0.500000
U           : 1.000000 0.000000 1.000000
V           : 1.000000 0.500000 0.000000
W           : 1.000000 0.500000 0.500000
X           : 1.000000 0.500000 1.000000
Y           : 1.000000 1.000000 0.000000
Z           : 1.000000 1.000000 0.500000
_           : 1.000000 1.000000 1.000000
```

TABLE 10.3: VECTOR MAP: TEXTUAL VISUALISATION.

When the user requests a snapshot during a process, an anchored view is connected to this process. As long as the user does not explicitly request disconnection, the evolving relations are shown. A request for a snapshot after a process can be considered as a visualisation of the final state in that process. During and after the training process, the state of self-organisation for each pair of dimensions can be examined. Here, the relations between neighbouring neurones are visualised as co-ordinates in the vector space. Also, the interneuronal distances can be visualised as can be seen in the following figure.

The fragment in Figure 10.4 shows the feature map from another, more static perspective, where cluster boundaries can easily be traced. The Euclidean distances between neighbouring neurones are presented as thickness degrees in a static grid.

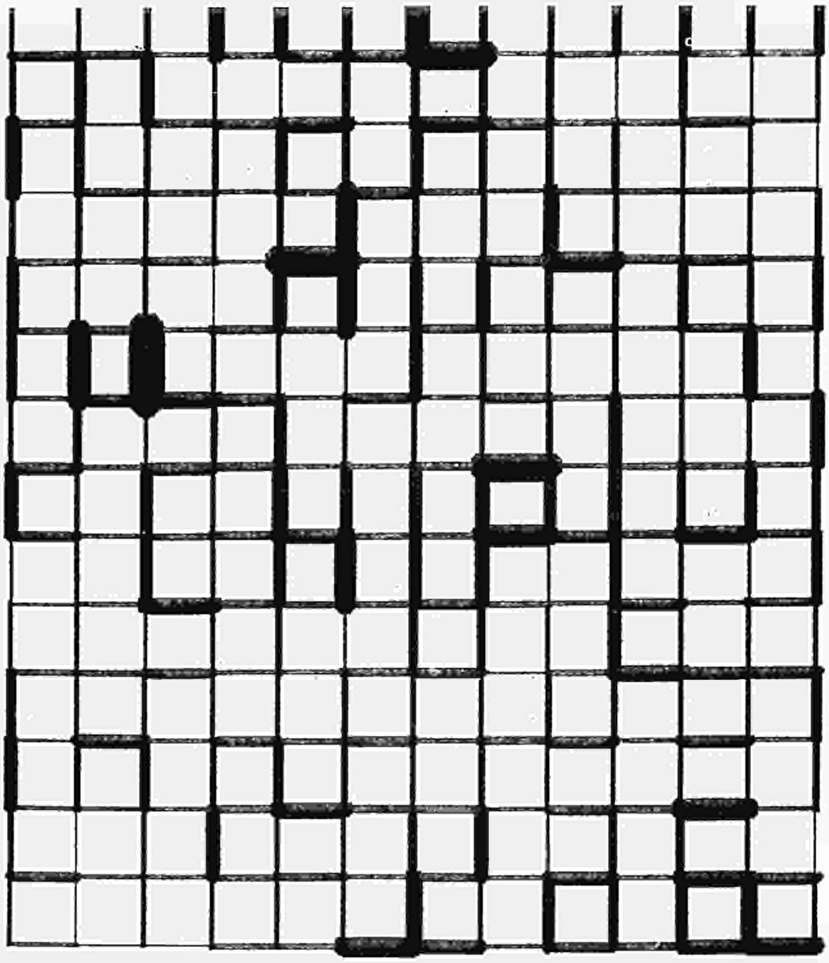
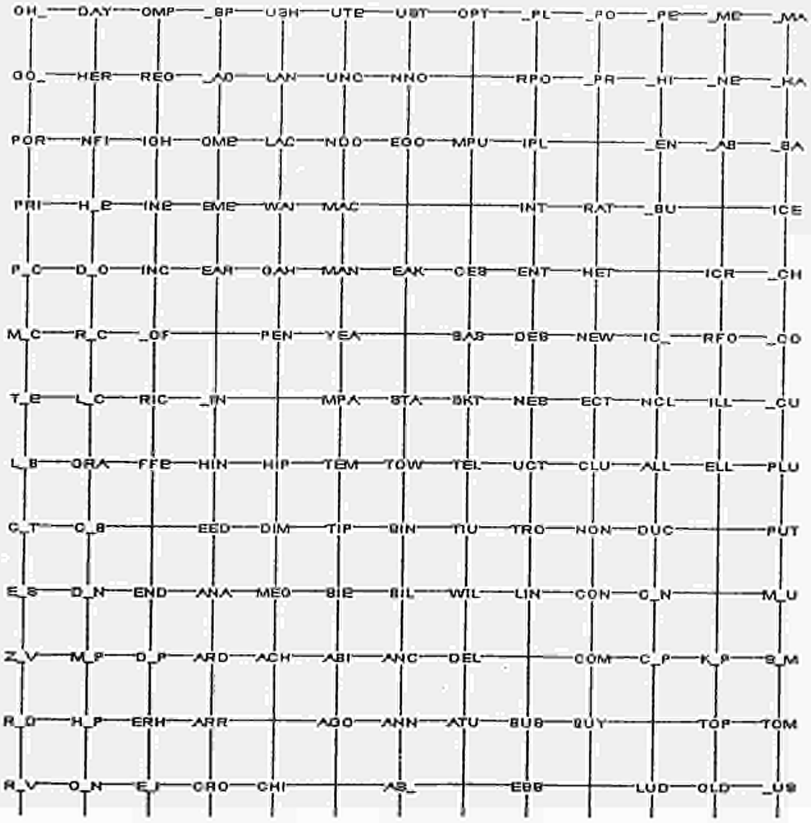


FIGURE 10.4: INTERNEURONAL DISTANCES VISUALISATION.

Another perspective is offered by the object distribution. The objects, or n-grams, can be examined from two perspectives. The first perspective visualises the best objects possible for each neurone (see Figure 10.5). The visualisation of the distribution of the best objects in the query requires a statistical frequency analysis[38] in the pre-processing phase. This results in an ideal object distribution, which can be useful in a comparison with the actual object distribution. To optimise comparisons, the actual object distribution visualisation includes a search mechanism where any object can be entered and its best matching unit is returned.



OH_	DAY	OMP	_SP	USH	UTE	UST	OPT	_PL	_PO	_PE	_ME	_MA
GO_	HER	REG	_AD	LAN	UNC	NNO		RPO	_PR	_HI	_NE	_KA
POR	NFI	IGH	OME	LAC	NOO	EGO	MPU	IPL		_EN	_AB	_BA
PRI	H_E	INE	EME	WAI	MAC			INT	RAT	_BU		ICE
P_C	D_O	INC	EAR	GAH	MAN	EAK	CES	ENT	HET		ICR	_CH
M_C	R_C	_OF		PEN	YEA		SAS	DES	NEW	IC_	RFO	_CO
T_E	L_C	RIC	_IN		MPA	STA	SKT	NES	ECT	NCL	ILL	_CU
L_S	ORA	FFE	HIN	HIP	TEM	TOW	TEL	UCT	CLU	ALL	ELL	PLU
C_T	C_S		EED	DIM	TIP	SIN	TIU	TRO	NON	DUC		PUT
E_S	D_N	END	ANA	MEG	SIE	SIL	WIL	LIN	CON	C_N		M_U
Z_V	M_P	D_P	ARD	ACH	ABI	ANC	DEL		COM	C_P	K_P	S_M
R_O	H_P	ERH	ARR		AGO	ANN	ATU	BUB	BUY		TOP	TOM
R_V	O_N	E_I	CRO	CHI		AS_		EBB		LUD	OLD	_US

FIGURE 10.5: OBJECT DISTRIBUTION (BEST OBJECTS IN QUERY - INCOMPLETE).

The empty co-ordinates are due to the fact that some patterns have the same neurone as their BMU. Only the most frequent object is printed in that case.

[38] In the implementation, a dynamic B-tree has been used to minimize execution time. The table map could not be used, because of its inefficient nature in the case a high context size is used. The B-tree object has also been applied to the common words to maximise the overall performance of the prototype.

Long term development of the map can be examined by visualising the error recording (see Figure 10.6). This object contains the Euclidean distances for each best matching unit for each pattern in the query and is visualised in a graph. The activity recording (see Figure 10.7) contains the complements of the Euclidean distances for each best matching unit for the 500 most recent patterns in the passing data. It is visualised in a graph, in relation to the hit threshold as well as the view threshold.

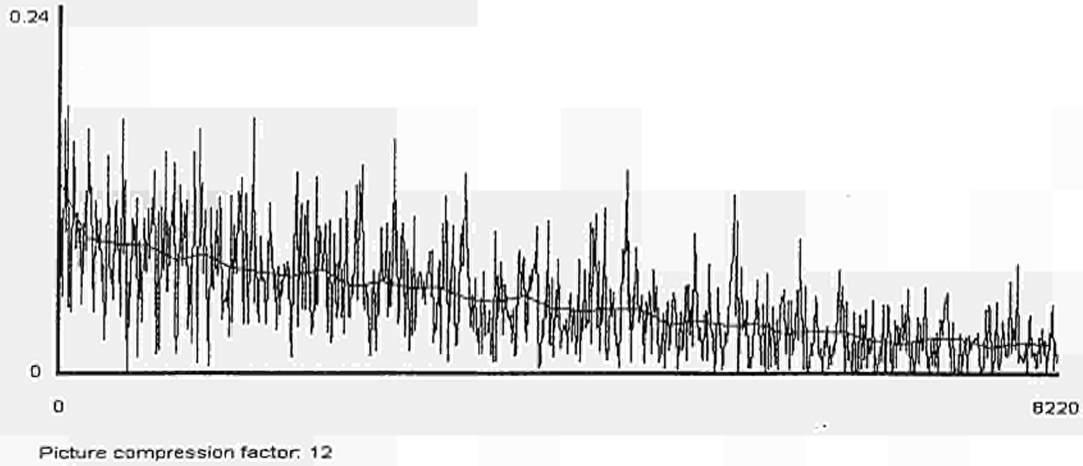


FIGURE 10.6: ERROR RECORDING.

The X-axis represents the maximum range of the pattern's Euclidean distances to their BMU's during the training process. The Y-axis represents the number of training cycles. In the evaluation, the queries have been presented ten times to the feature map. As the feature map converges to its final representation, the overall errors get smaller. This means that the feature map finds an overall way to represent the query. Peeks are due to infrequent patterns. Picture compression means that only one per twelve errors is plotted to give a overall impression.

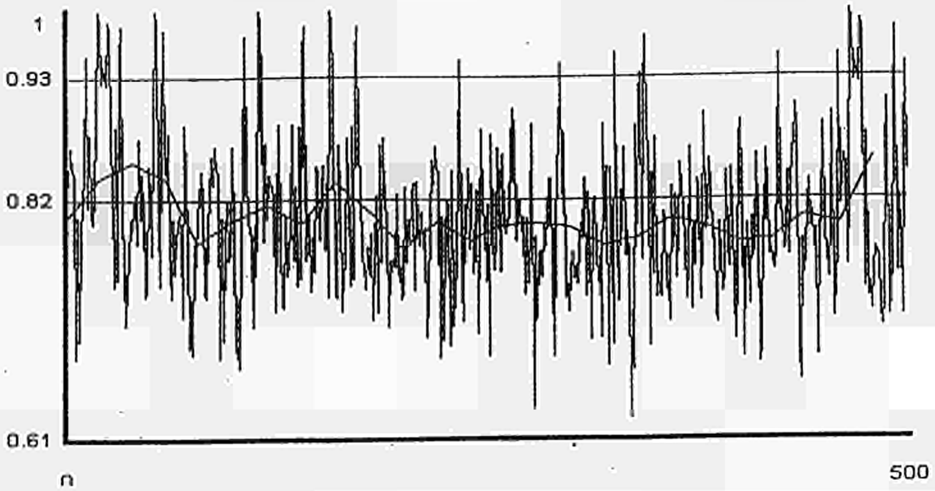


FIGURE 10.7: ACTIVITY RECORDING.

The X-axis represents the maximum range of the complement of the pattern's Euclidean distances to their BMU's during the extraction process. The Y-axis represents the most recent extraction cycles. Here, the upper boundary (at 0.93) represents the perfect hit threshold. The lower boundary (at 0.82) represents the view threshold.

Accessibility

The concept of accessibility stresses the importance of storage and retrieval of all input and output to enable reconstructions and variations of experiments. Therefore, the prototype supports three document-view formats, which are invoked by a open/save-command:

- FILTER Output (*.out / *.txt), i.e. the standard ASCII text format, used for textual visualisations, the system report and external data.
- FILTER Picture (*.pic), i.e. a special format which supports system- and user drawing, as well as dynamically sizeable text. This format is used for graphical visualisations.
- FILTER Hitlist (*.hit), i.e. a special database format which uses Open DataBase Connectivity (ODBC) to connect only to external DBase (*.dbf) databases with the prototype database structure.

The prototype also supports four document-object formats, which are invoked by a load/save-command:

- FILTER Settings (*.set), i.e. a special, protected format which contains the configuration of parameters, preferences, files and paths and system settings.
- FILTER Demo (*.dem), i.e. a derivation of the settings format which also contains the special demo settings, which causes the demo mechanism to use artificial vectors instead of natural language data. This is useful for simulations of ideal situations.
- FILTER Data (*.dat), i.e. a special ASCII format which contains the contents of all objects, together with their dimensions, except the table map (because of its optional nature).
- FILTER Table (*.tbl) , i.e. a special ASCII format which contains the table map with its dimensions.

With these formats, the prototype does provide maximum accessibility. Especially the FILTER Hitlist format is of great importance, because it enables each user to optionally decide the amount of returned information by merely resetting the view threshold.

A Session with the FILTER Prototype

To conclude this section, a global overview of the prototype will be given in the form of an imaginary session.

When the session is started, a workspace appears, containing eight menus, a toolbar, a status bar and a system log-file. This log-file reports that the most recently used settings file has already been reloaded.

If this session is to be a continuation of the last session, only the corresponding data has to be loaded. After loading the data, all menu commands become accessible, indicating that the pre-processing phase, i.e. all processing needed to start the extraction process, has been completed.

However, if this session is not to be a continuation of the last session or if there is no data file which corresponds with these settings, the complete process cycle has to be pursued. To start with, the settings which correspond best to the current aim should be loaded. To alter these settings, the Parameters dialogue, the Preferences dialogue and the Files and Paths dialogue can be opened to reconfigure these settings. Within the Parameters dialogue, the Hints dialogue can also be opened, which can be of great help to optimise the settings. Having fine-tuned this configuration, these new settings ought to be saved.

Now the system has been configured, the actual process can be started. First, the data objects have to be initialised according to these settings. Next, the query must be taught to the feature map. Depending on what visualisation preferences have been set, a number of new windows appear, which offer views from different perspectives onto the training process. These visualisation preferences, like all settings, can be altered at any moment. For example, if the representation process does not seem to work out well, it is possible to interfere by adjusting training parameters like epsilon, i.e. the learn rate, and sigma, i.e. the region update area. Of course, the process can also be interrupted.

When the training process ends, the system asks whether a table map for this feature map must be generated. If a low context size has been used, this question should be answered affirmative. After the table map generation, saving the data is necessary to be able to reuse these data objects in future sessions. The system asks whether the table map should also be saved.

At this point, the pre-processing phase has been concluded. All commands are accessible now, including the Extract Text Parts command. When the extract process is started, three more new windows appear: the internal data flow view, the feature map activity view and the hitlist view. The first two views can be inactive, depending on the preferences settings. The hitlist view cannot be deactivated. When this view has the focus, the hitlist-navigation toolbar buttons become activated to provide a convenient way to also browse through the database as the hitlist is being build. When the contents preview, contained within the hitlist view, looks interesting, the Retrieve button in the view can be pushed to examine the whole text part in a separate view. The hitlist can also be ordered on one of the five available fields during this process. To fine-tune retrieval, the view threshold can be adjusted to increase or decrease the number of retrieved documents. This can best be done after examining the activity view, because it shows the activity in relation to the thresholds. To activate a new view threshold, the sort hitlist-command must be called to update the database.

Once the extraction process has ended, the hitlist ought to be saved to enable future access. Now the hitlist can be printed, reordered, edited and reviewed at all times.

The imaginary session described here, assumes the user is willing to experiment somewhat within this neural filtering environment. If this is not the case however, the process cycle can be minimised with respect to the user effort by activating the evaluation mode.

In this evaluation mode, all the user has to do is prepare a number of settings, select those settings in the Evaluation dialogue and press the Evaluate button. This mode implicitly saves all data and hitlists. In this mode, the user can simply do something else on the computer, because the prototype also supports smooth background processing.

10.3 Evaluation

This section explains how the prototype, as described in the previous section, has been evaluated. First it describes the preparation phase, i.e. how the data set was composed, how the queries were selected, how the settings, or the parameter configurations, were set. Then it is described how the correlation, or the degree of resemblance, between the query representation in the feature map and each document in the data set was calculated.

After that we describe the execution & analysis phase, starting with a detailed report on the preliminary outcomes. Based on these preliminary outcomes, additional tests and analyses have been carried out, which are also described in detail.

After that a comparison is made between the FILTER prototype and ZyIMAGE, a contemporary index-based information retrieval system.

Preparation

Data set

The data set consisted of 100 image-based, rather accurately scanned, articles (823 KB) from *the Wall Street Journal Europe, August 8-18, 1994.* The article collection was composed by querying ZyIMAGE, an index and image based document retrieval application. Three elementary queries or interest profiles were used: WAR*, EC and COMPUT* . The inclusion of wild cards ensured that the recall would be high and the precision low . In theory, there should have been three topics in this data set, one for each query. In practice however, there was only one coherent group: the COMPUT*-group. This is partly because this group was composed out of documents with a relatively high hit density only. Another factor could be that words beginning with the substring WAR have too diverse semantics, whereas the string EC is too specific. Words beginning with the substring COMPUT all seem to belong to the same semantic class “Computer terms”. The composition of the data sets for the three interest profiles can be seen in Tables 10.4 a,b,c,d.

TABLE 10.4 A: DATA SET COMPOSITION. QUERY 1: WAR* (NO FUZZY SEARCH WAS USED) COMPLETE RETRIEVAL, PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
W1	QJ.TXT	15	Warburg, investment bank, hostile takeover
W2	FZ.TXT	2	federation of taxpayers, warns, tax burden taxpayers
W3*	13V.TXT	7	warnings, Compaq, computer keyboards, wrist injuries
W4	G5.TXT	4	Time Warner inc, Viacom inc. sells theater chain
W5	1F6.TXT	2	fare war, eurotunnel denies fare reductions
W6	10I.TXT	23	marketing & media, iced tea to europe, warner bros. records
W7	10G.TXT	6	Unilever, Omo power, detergent war

W8	AW.TXT	6	Cisco systems, communicating with small investors
W9	X8.TXT	4	pharmaceutical industry, takeover, Glaxo holdings
W10*	13W.TXT	2	IBM, order system for software
W11	18K.TXT	1	Alliance Pharmaceutical, Johnson & Johnson, drugs group
W12	10H.TXT	4	Unilever, Omo power, Proctor & Gamble, soap war
W13	PX.TXT	3	Unilever pretax profit rose
W14	13O.TXT	1	wholesale prices fall in western Germany
W15	JM.TXT	4	car trip in Europe, reasons to stay at home
W16	AJ.TXT	6	business opportunity in Poland, financing
W17	AK.TXT	1	Russia plans to take action on plague of unpaid bills
W18	9T.TXT	5	Murdoch's price war, newspaper sales up, profits are falling
W19	Q7.TXT	3	World Bank warned Turkey, international loans and credits
W20	G2.TXT	2	Polish court, license, PolSat TV, nationwide broadcasting
W21	X3.TXT	2	U.S. industrial production +0.2% in July, warm weather
W22	70.TXT	1	Britain, Smart-card technology, drivers, EC directive
W23	9X.TXT	3	British monetary-policy, industrial production, Warburg
W24*	1HI.TXT	2	Hewlett-Packard share price rises on increases in earnings
W25	Q1.TXT	2	microwave filter, air warfare, communications applications
W26	10K.TXT	2	M6, French television, bourse listing
W27	FP.TXT	3	German teens find new fuel for disco raves, Red Bull
W28	FS.TXT	1	Kuwait seals Russian ties with major arms purchase
W29	AZ.TXT	2	China's elusive effect on market for commodities
W30	AE.TXT	2	U.S. FCC, license, interactive television services
W31	D0.TXT	1	ING Group, Bank Brussels Lambert, takeover
W32	CL.TXT	2	British Airways, rise in pretax profit, shares fall
W33	CK.TXT	1	Lending in U.K. to consumers rises to record
W34	AL.TXT	2	AIDS, epidemic, research, conference
W35	6K.TXT	2	CNN, BCC, Cox, 24-hour news business, competitors
W36	PM.TXT	2	NATO begins search for new top secretary-general
W37	PS.TXT	1	TBB, CAA, pact, video programming, telephone customers
W38	73.TXT	2	German postal service will offer 25% of shares to public
W39	9S.TXT	2	Europe's economic recovery is beating expectations
W40	JN.TXT	2	Zurich, Fust Investors, takeover, insider trading
W41	71.TXT	1	costly diet products in Russia, Herbalife, scientific?
W42	CR.TXT	1	Saatchi & Saatchi Co., profit, British advertising, marketing
W43	1HD.TXT	3	American Express, data mining, Sybase Inc
W44*	A0.TXT	1	IBM's overhaul of disk-drive unit may cut jobs in Europe
W45	AQ.TXT	1	Union Bank of Switzerland, shares drop 28%
W46	A1.TXT	1	Finalists to purchase Kodak's household-products division
W47	CX.TXT	3	China should take a hard line on software pirates
W48	CH.TXT	1	Italian Silvio Berlusconi, television advertising, RAI
W49	GD.TXT	1	German teens find new fuel for discos, continued
W50	XL.TXT	2	Toys R Us, investors, Petrie Stores, deal, Warburg complex
W51	146.TXT	2	new markets in central and eastern europe, investors
W52	CQ.TXT	1	Germany, Bayer AG, buying drug business of Kodak
W53	6Y.TXT	1	global news business, BBC challenges CNN
W54	AR.TXT	1	European central banks, Germany's Bundesbank, discount
W55	1JN.TXT	2	Russian mafia, new problems, La Cosa Nostra
W56	13P.TXT	1	American Home Products buys American Cyanamid
W57	10B.TXT	1	German drug investigation, two Schering AG drugs
W58	X2.TXT	1	Portuguese Bank at war, free-market policies
W59	QA.TXT	1	Algeria, nationalist guerrillas, Islamic slogans, war
W60	WZ.TXT	1	Carlos the Jackal, arrest, terrorism, politics, Sudan, France
W61	FQ.TXT	1	French cognac, consumption dropped, Norwegians buy cars
W62	X1.TXT	1	trends, business travel and tourism, economic recovery
W63	PL.TXT	1	Maastricht, fiscal policy, Europe, monetary union
W64	6H.TXT	1	scandals, Washington, Congress, lobbyist-paid trips
W65	1JU.TXT	1	LDDS Communications inc., telephone, acquisition, WilTel
W66	10A.TXT	1	Russia, Germany, plutonium smuggling, pressure
W67*	CG.TXT	2	Reebok, workers' rights, China, software, aquarium, fish
W68	6S.TXT	1	Ted Turner, New Line Cinema, Hollywood, movie business
W69**	A3.TXT	1	Dell Computers, return, notebook, new designs, price battle
W70	13S.TXT	1	Scandinavian Airlines System, recovery, pretax profit
W71	G4.TXT	1	record stores, changing, multimedia stores
W72	G7.TXT	1	cars, Renault SA, front-runner in French privatization race

W73	1HG.TXT	1	U.S. retailing, Dayton Hudson, Wal-Mart Stores, earnings
W74	1JO.TXT	1	Johnson & Johnson acquires Neutrogena, personal care
W75	1IG.TXT	1	U.S. health care, problem,subsidized by federal government
W76	13G.TXT	1	Vietnam places hope for economic health in local enterprise
W77	6O.TXT	1	American Cyanamid, American Home Products, takeover
W78	JK.TXT	1	Japanese price revolution, consumer behaviour, shopping
W79	13U.TXT	1	technology & health, employees, virtual offices, low morale
W80	GA.TXT	2	Boston sees payoff and problems in east Europe, expanding
W81	B1.TXT	1	Asian markets, bourses consolidate after gains, Tokyo
W82	PJ.TXT	1	Bill Clinton, defeat on Crime Bill, Washington, Congress
W83	1HU.TXT	1	Helsinki, Oy Nokia, toilet-paper, cellular phones
W84	G3.TXT	1	British, high taxes, bargain hunters dent U.K. alcohol sales
W85	B0.TXT	1	Eurobond Market, quiet week
W86	GK.TXT	2	international bond indexes, bund prices fall

TABLE 10.4 B: DATA SET COMPOSITION. QUERY 2: EC (NO FUZZY SEARCH WAS USED) COMPLETE RETRIEVAL, PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
E1	13J.TXT	1	EC lobbyists, American Express, Brussels
E2	FY.TXT	1	Terra Industries buys fertilizer products concern (NO ec!)
E3	10I.TXT	1	marketing & media, iced tea to europe, warner bros. records

TABLE 10.4 C: DATA SET COMPOSITION. QUERY 3: COMPUT * (NO FUZZY SEARCH WAS USED). INCOMPLETE RETRIEVAL (THE 18 HIGHEST RANKED DOCUMENTS OUT OF 50 ARE SELECTED), PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
C1**	1F7.TXT	6	Dell Computers, overhaul, desktop computers,Pentium,Intel
C2**	A4.TXT	7	Compaq's flagship line of notebook computers, defect
C3*	13V.TXT	10	warnings, Compaq, computer keyboards, wrist injuries
C4*	13W.TXT	6	IBM, order system for software
C5**	1QF.TXT	9	IBM plans to slash prices to counter Compaq, overhaul of PC line
C6*	10F.TXT	5	bidding for Ziff Communications, computer magazines
C7*	XA.TXT	7	European PC sales gained, Compaq, IBM, Apple
C8**	A3.TXT	7	Dell Computers, return, notebook, new designs, price battle
C9	AD.TXT	1	Lufthansa AG, freight, maintenance, computer operations
C10	1QD.TXT	1	recognition factor, Swiss Bank, note-counting, manual labor
C11*	1QW.TXT	4	AT&T, Intel, software methods, PC-Based, videos
C12	108.TXT	1	remote answering-machine service, computer technology
C13	XJ.TXT	6	technophobia, human skills vs. information technology
C14*	1HI.TXT	3	Hewlett-Packard share price rises on increases in earnings
C15*	1QV.TXT	5	detente between Novell and Microsoft, product tuning
C16*	A0.TXT	3	IBM's overhaul of disk-drive unit may cut jobs in Europe
C17*	1F9.TXT	1	CompUSA Inc., struggling U.S. computer-superstore chain

TABLE 10.4 C: DATA SET COMPOSITION: NOTES

A number of documents have multiple occurrences

- 13V.TXT* : W3 ⇔ C3.
- 10I.TXT : W6 ⇔ E3.
- 13W.TXT*: W10 ⇔ C4.
- 1HI.TXT* : W24 ⇔ C14.
- A0.TXT* : W44 ⇔ C16.
- A3.TXT** : W69 ⇔ C8.

** means the document is closely related to the query C1. These 4 articles only are directly about computer manufacturers like Dell Computers and about one of their computer models. These articles must therefore be extracted.

* means the document is somewhat related to the query C1. These 10 articles are about computer-related companies. These articles may therefore be extracted.

Queries

Testing has been carried out with two types of queries:

- A *full-text query,* i.e. a document in natural language.
- An *artificial query,* i.e. a concatenation of keywords, separated by a non-character.

As the full-text query, the highest ranked document of the C-group (C1 in the above table) was chosen. By choosing a document from the data set as the query, the maximum activity for the neural net gets implicitly defined. This has been used to determine the relative activity of all other documents as well as to optimise the hit threshold. Taking the document with the highest hit density ensures that its dominant patterns, or features[39], are represented in the map after the training process.

The artificial query was composed by concatenating all informative words in the full-text query, separated by a comma. By deriving the artificial query from the full-text query a comparison between the results can be made. One advantage of the artificial query could be that, because of it is cleaned up from noise, a better representation can be formed on the feature map after the training process. A practical advantage is the smaller size of the neural net needed to represent the query, simply because the query is much smaller. This speeds up the overall performance of the system. Also a comparison with index-based information retrieval systems can be made, due to the artificial query's resemblance to a weighted quorum query.

The text of the full-text and the artificial query can be found in Table 10.5.

TABLE 10.5: FULL TEXT QUERY

```
C1 (1F7.TXT):
--------------------------------------------------------------------------------
<scan_date> 8/22/94 </scan_date>
<source> Wall Street Journal Europe </source>
<title> NA </title>
<author> NA </author>
<copyrights> NA </copyrights>
<abstract> NA </abstract>
Dell Plans to Overhaul Desktop Computers Aimed at Companies
By a Staff Reporter
AUSTIN, Texas - Den Computer Corp. is expected to announce today an overhaul of its high-end OptiPlex line of corporate
desktop computers that will include price cuts and the introduction of high-perform- ance Pentium microprocessors. The computer
vendor said it would break,a price barrier by offering for under $3,000 a fully configured desktop system based on a speedy 90-
megahertz version of Intel Corp.'s Pentium chip. "We are moving toward Pentium carry- ing over into the corporate side," said
```

[39] In the case of this query, in combination with trigrams, examples of features are COM, OMP, MPU, etc.

Doug MacGregor, Dell's vice president for desktop computers. "None of our corporate customers have a question of whether or not they'll use Pentium. It's a question of when." Dell has pushed hard with the newest Intel chip, and now more than half of all Dimension machines sold to home users and small businesses use Pentium chips, Mr. MacGregor said. But corporate customers have been waiting for Pentium prices to fall, he said. With the new Optiplex prices, businesses will be able to buy Pentium-based machines at prices similar to what they were paying less than a year ago for slower 486-based computers, Mr. MacGregor said. Dell said its new OptiPlex models replace machines introduced about one year ago. The new machines incorporate ad- vanced power management, enhanced net- working capabilities and easier-to-use "plug In' play" features.

```
<xref image="J:\INDEX\ALGEMEEN\TIFF\1F7_01.tif|0"> image: </xref>
--------------------------------------------------------------------------------
```

TABLE 10.6: ARTIFICIAL QUERY

```
C1', derived from C1 (1F7.TXT):
--------------------------------------------------------------------------------
Dell,Desktop,Computers,OptiPlex line,
desktop computers,introduction,high-performance,
Pentium microprocessors,computer,price barrier,
configured desktop system,speed,megahertz,
Intel,Pentium,chip,Pentium,Doug MacGregor,Dell,
desktop computers,Pentium,Dell,Intel chip,
Pentium chips,MacGregor,Pentium,Optiplex,
Pentium-based machines,486-based computers,
OptiPlex models,power management,networking,
"plug In' play"
```

Settings

Four parameters have been exhaustively tested:

- The generalisation factor, i.e. the ratio of the network dimensions to the number of n-grams in the query. Values of 2, 4 & 6 have been processed.

- The context size, i.e. the size of the window which is being shifted over the data. Values of 3, 5 & 7 have been processed.

- The space as character, i.e. the usage of the space character to include a word's natural context better in the representation in the feature map. Values of 0 & 1 have been processed. Note that this parameter will not be varied when the artificial query is processed. In that case there is *by definition* no natural context.

- The hit threshold, i.e. the degree of correlation a n-gram must have with the best matching neurone in the feature map to be recorded as a perfect hit. This is essential in the extraction process, if the hitlist is to be sorted on the perfect hit rate or the average hit error. Although this parameter could also be altered during or after the actual extraction process, this would of course make the perfect hit rate and the average hit error not reliable anymore. Therefore, in the preliminary evaluation, the hit threshold has been kept constant at a value of 0.2. In the additional evaluation the hit threshold will be varied, based on the preliminary results.

Note that this means there has not been looked in detail into the form of the feature map, the learning rate and the weights-update region size during the training process, possible optimisations of vector-character assignments, and so on. All these parameters have been kept constant on values, based on experimental pre-processing as well as on former research by others.

Definitions of Correlation

Three levels of *correlation* were determined manually for all documents in the data set: extreme correlation, significant correlation and no correlation. Four documents **had to be** retrieved to achieve a maximum recall. Only these documents are directly about computer manufacturers *and* about one of their computer models. Ten other documents were **allowed to be** retrieved to continue a maximum precision. These documents are about computer-related companies or about computer-related products. All other 86 documents were rated irrelevant (see also the notes below the tables with the data sets)

The term information value has been used as a reversed value of a pattern's probability of occurrence. This means that if the probability a pattern will occur is high, its information value is low.

The term precision has been used to indicate the number of correlating documents which were retrieved before an irrelevant document was returned.

The term recall has been used to indicate the number of documents which were retrieved before all four extremely correlated documents were returned.

The term *accuracy* has been used as an extension of recall. It has only been used where the recall-value and the precision-value were identical. Therefore it includes a measure of quality. Traditionally speaking, it is the precision at 100% recall. This term has been used as principal retrieval measure instead of the traditional terms, because in a real filtering situation the one disastrous event for a user is missing valuable information and not be able to know it. Therefore, the user must be offered an ordered database. This way, personal view thresholds can be set, making it possible to always retrieve all relevant information. Because of this view, precision and recall ratios become less relevant as retrieval measurements. The user expects 100% recall. At what position in the ranking this is achieved, should in the end be for the user to decide. This approach to retrieval measurement is captured in the term accuracy.

Correlation calculation

As measures of correlation, three values have been calculated to determine the optimum Precision and Recall ratio:

- The average error, i.e. the cumulative Euclidean distance to the best matching unit in the neural net for all n-grams in the document, divided by the number of n-grams in the document. This can be thought of as a negative filter, for correlation in this concept is a result of distance calculations. Its value is composed out of information of all n-grams, which results in a global measurement of a document.
- The perfect hit rate, i.e. the cumulative number of n-grams in the document of which its Euclidean distance to the best matching unit is smaller than the hit threshold, divided by the number of n-grams in the document. This can be thought of as a positive filter, for correlation in this concept is a result of counting hits. Its value is not composed out of information of all n-grams, which can result in a global as well as a local measurement of a document. If, for example, only one section in a document correlates to the feature map representation, the document can still be retrieved.
- The average hit error, i.e. the cumulative Euclidean distance to the best matching unit in the neural net for all n-grams in the document of which its Euclidean distance to the best matching unit is smaller than the hit threshold, divided by the number of n-grams in the document. This can be thought of as a positive-negative filter, for it is a fusion of a positive and a negative filter. Correlation in this concept is a result of valuating hits by distance calculations. As in the positive filter, its value is not composed out of information of all n-grams, which can result in a global as well as a local measurement of a document. In the example at b), this positive-negative filter could also clarify how well that section of that document correlates to the feature map.

Expectations

After training, the representation of the full-text query on the feature map was expected to be less accurate than the artificial query's representation, because the full-text query contains a lot of noise, even after passing the input filter. Therefore, the full-text query was expected to be less accurate in the extraction process as well. [artificial +, full-text -]

If the generalisation factor becomes too high, the query's discriminating features will fade too much. Then no accurate correlation between the data flow and the query representation can be distinguished anymore. [generalisation factor +, accuracy -]

With the context size, the information value of a pattern increases exponentially. In other words, the patterns in the feature map as well the patterns of the data flow are more distinctive when the value of the context size is high. [context size +, accuracy +]

In the case of the full-text query, the inclusion of the space as a character to incorporate word adjacencies results in a much higher number of possible patterns in the data stream, while the number of actual patterns in the query does not increase that much, because relatively few combinations appear in the query. This will decrease the perfect hit-probability and thus increase a perfect hit's information value. Therefore, distinguishability increases between relevant and non-relevant documents. [space +, accuracy +]

The average error calculation should serve as a general indication of document relevancy, because of its insensitivity to perfect hits. The perfect hit rate and the average hit error are possible optimisation options, which should at least do well in the case of the full-text query with inclusion of the space as a character.

Execution of the evaluation experiments

The evaluation of the FILTER prototype has been conducted in two stages. First, many different parameter settings were tried in a semi-random search fashion. The goal of this phase was to investigate the effects of the parameters on the behaviour of the system. The results obtained in this phase are described under the heading "Preliminary Results". Second, the more promising regions of the parameter space were isolated and experiments were conducted to get peak performance out of the system. These results are described under the heading "Additional Results".

Preliminary Results

In this section the preliminary results table (Table 10.7 a,b) is globally reviewed, from left to right and from top to bottom. For a good understanding of this table, it should be noted that there are 14 documents which may be retrieved to continue the state of maximum precision: C1, C2, C5, C8, C3, C4, C6, C7, C11, C14, C15, C16, C17, W67. The first 4 documents, however, must be retrieved to reach the state of maximum recall.

TABLE 10.7 A: PRELIMINARY RESULTS: TABLE ABBREVIATIONS.

Settings:	
FTQ	= Full Text Query
AQ	= Artificial Query
Digit 1	= Generalisation factor
Digit 2	= Context size
Digit 3	= Space as character
Digit 4,5,6= Hit threshold (its float value only)	
Q. error:	Average error of the query itself, i.e. the complement of the maximum map activity
D. error:	Average error of the best matching document in the data set
P*x*:	Maximum Precision document count with hitlist sorted on *x*
R*x*:	Maximum Recall document count with hitlist sorted on *x*
E*x*:	Error document count at maximum recall with hitlist sorted on *x*
*x*1:	Count with hitlist sorted on Average error
*x*2:	Count with hitlist sorted on Perfect hit rate
*x*3:	Count with hitlist sorted on Average hit error

TABLE 10.7 B: PRELIMINARY RESULTS TABLE.

Settings	Q. error	D. error	P1	R1	E1	P2	R2	E2	P3	R3	E3
FTQ23020	0.119695	0.177454	1	11	2	2	13	6	1	80	70
FTQ43020	0.159510	0.206315	2	14	6	4	18	9	1	37	26
FTQ63020	0.181064	0.217626	1	11	2	1	11	4	1	18	11
FTQ25020	0.179135	0.259371	1	13	3	5	10	2	3	9	3
FTQ45020	0.215698	0.275734	2	14	4	4	13	5	2	15	6
FTQ65020	0.234187	0.287028	1	14	4	4	8	1	1	10	3
FTQ27020	0.219438	0.301106	1	11	3	*6*	*6*	*0*	*5*	*5*	*0*
FTQ47020	0.249470	0.310746	1	15	5	*7*	*7*	*0*	*7*	*7*	*0*
FTQ67020	0.268693	0.318401	1	14	5	*6*	*6*	*0*	*6*	*6*	*0*
FTQ23120	0.120582	0.168126	6	6	0	2	14	5	1	70	59
FTQ43120	0.155662	0.200007	3	8	1	1	15	8	1	34	25
FTQ63120	0.182588	0.217062	2	11	2	1	21	12	1	34	25
FTQ25120	0.183954	0.252867	3	7	1	6	8	1	3	10	4
FTQ45120	0.218768	0.273276	4	8	2	7	7	0	2	6	1
FTQ65120	0.240167	0.286132	3	7	1	4	9	1	2	8	1
FTQ27120	0.228325	0.300309	4	8	2	8	8	0	5	8	1
FTQ47120	0.257845	0.312278	4	8	2	6	6	0	6	6	0
FTQ67120	0.276522	0.321395	4	10	3	6	6	0	6	6	0
AQ23020	0.202122	0.226122	4	15	5	2	28	16	2	58	45
AQ43020	0.232747	0.249897	3	18	9	4	28	16	4	56	43
AQ63020	0.253562	0.267204	2	30	18	2	20	9	1	30	18
AQ25020	0.268683	0.295165	3	12	3	7	7	0	6	8	1
AQ45020	0.284676	0.308447	3	14	5	6	6	0	4	8	2
AQ65020	0.299796	0.320611	2	25	15	8	8	0	5	8	1
AQ27020	0.296442	0.326402	2	26	15	7	7	0	7	7	0
AQ47020	0.315326	0.337489	1	30	19	8	10	1	7	11`	2
AQ67020	0.328301	0.344352	2	37	25	6	13	3	5	16	6

Figure 10.8 presents the precision and recall graph for the full-text query with space as a character. Only the accurate results are visualised here. The actual accuracy value is contained within these charts at the point the fourth and last highly relevant document, i.e. *n*=4, has been retrieved.

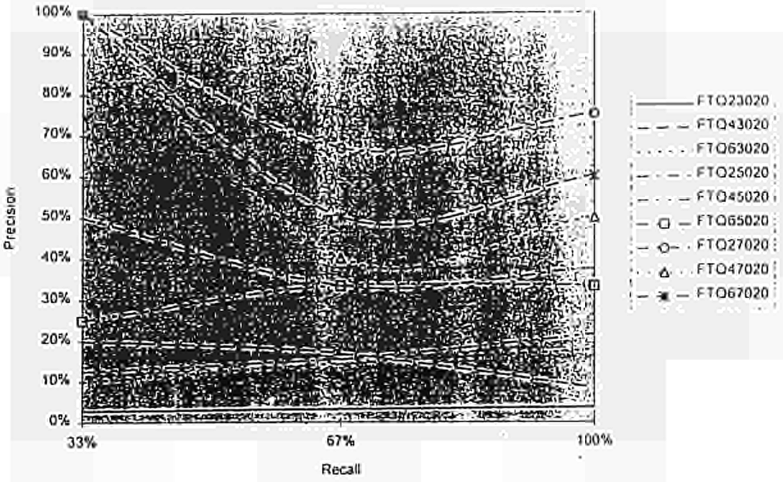


FIGURE 10.8: PRECISION VERSUS RECALL (FULL-TEXT QUERY, SPACE AS A CHARACTER, SORTED ON AVERAGE HIT ERROR) - PRELIMINARY RESULTS:

The difference between the average error of the query and the average error of the best matching document is significantly smaller in the case of the artificial query than in the case of the full-text query. Whenever extremely accurate hit threshold sensitive results were returned, the average error of the query was always higher than the hit threshold.

When sorted on the average error, the full-text query without space as a character did not do so well. Even its best results were not acceptable. The full-text query with space as a character did do better. The results even turned out to be very accurate, when the generalisation factor as well as the context size were set low. The artificial query did not do well at all.

When sorted on the perfect hit rate, the full-text query without space as a character did do very well when, but only when, the context size was set high. In these settings a state of extreme accuracy was reached. The generalisation factor had become insignificant at this point. The full-text query, with space as a character, did even do slightly better. In these settings the state of extreme accuracy was already more or less reached using only a medium context size, but at this point the generalisation still played a significant role. The artificial query also became extremely accurate when the context size was set medium. With a high context size, the accuracy decreased again. At this point the generalisation factor became relevant again.

When sorted on the average hit error, the results were also extremely good in some configurations, but in general less stable than the results for the perfect hit rate. These two hit threshold-sensitive correlation calculations seem to react in the same way to parameter variations, but the perfect hit rate yields somewhat better results.

Preliminary Analysis

The artificial query is noiseless. Because of this property, any generalisation of its data will therefore reduce its discriminating features and thus decrease the accuracy of the query representation. This is reflected in the small difference between the average error of the query and the average error of the best matching document. This effect can be suppressed though, by using a higher context size. The information value of a query pattern increases more than it is decreased by generalisation. However, when the context size is set too high, too many keywords with a length, smaller than the context size, are incorrectly represented by the internal keyword concatenation.

The generalisation factor must, in general, not be set too high to avoid significant reduction of its discriminating features. However, this parameter is only of importance when the context size is not set high.

Without the inclusion of space as a character, the information value of the patterns in the full-text query can become too low. Also, by inclusion of the space as a character and thus the inclusion of adjacent word relations, the relative number of relevant patterns increases significantly[40]. Therefore, with the space as a character, the information value increases as well as the accuracy of the query's representation in the feature map.

The context size must not be set too high to avoid inaccurate representation of discriminating features[41]. However, if a good representation of only a few, but relevant, patterns has been formed in the feature map and the context size has been set high, the generalisation factor and the space as a character become relatively insignificant. Although the average error will be of a relatively indeterminate nature, the perfect hits these few patterns will cause, will be extremely informative *when the hit threshold is set lower than the maximum map activity and the hitlist is sorted on the perfect hit rate.*

[40] For example, the string _DESKTOP_COMPUTER_ is decomposed into 16 fully correct trigrams, whereas DESKTOPCOMPUTER is decomposed into only 11 fully correct trigrams.

[41] For the query used in this evaluation, discriminating features are for example DELL, CHIP, INTEL, etc.

The instability of the average hit error-sorted hitlists can be explained by the paradoxical nature of these values. The results are only accurate when the hit threshold is set relatively low. This means there are relatively few perfect hits. Thus the intentional effects are suppressed. In other words, this correlation calculation only performs well when it is transformed into a perfect hit rate imitation. Therefore, the average hit error-sorted results are considered not to be relevant.

To validate this analysis, new settings have been evaluated. There has been focused on configurations with a low context size to minimise execution times.

Additional Results

In this section the additional results table is globally reviewed, from top to bottom. The additional results can be found below in Table 10.8 (see Table 10.7 for the abbreviations used).

TABLE 10.8: ADDITIONAL RESULTS TABLE

Settings	Q. error	D. error	P1	R1	E1	P2	R2	E2	P3	R3	E3
FTQ27030	0.219438	0.301106	1	11	3	1	16	9	1	20	12
FTQ27010	*0.219438*	*0.301106*	1	11	3	*6*	*6*	*0*	*6*	*6*	*0*
FTQ23012	0.122862	0.180017	1	11	2	2	7	1	1	10	4
FTQ23009	0.122862	0.180017	1	11	2	2	9	1	1	10	3
FTQ23006	*0.122862*	*0.180017*	1	11	2	*6*	*6*	*0*	*7*	*7*	*0*
FTQ23109	*0.120582*	*0.168126*	*6*	*6*	*0*	*6*	*6*	*0*	4	7	1
FTQ23106	*0.120582*	*0.168126*	*6*	*6*	*0*	*4*	*4*	*0*	*4*	*4*	*0*
FTQ43107	0.155662	0.200007	3	8	1	1	23	11	1	36	24
FTQ43111	*0.155662*	*0.200007*	3	8	1	*6*	*6*	*0*	5	9	2
AQ13020	0.176395	0.205944	5	10	2	4	19	9	1	47	35
AQ130176	0.176395	0.205944	5	10	2	5	15	5	1	39	29
AQ130132	0.176395	0.205944	5	10	2	9	10	1	2	28	18
AQ130088	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	7	15	5
AQ130044	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	7	7	*0*
AQ130022	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	*8*	*8*	*0*
AQ15020	0.237150	0.279872	3	13	3	7	11	2	3	13	4
AQ230155	0.202122	0.226122	4	15	5	2	18	7	1	31	19
AQ230101	*0.202122*	*0.226122*	4	15	5	*9*	*9*	*0*	*9*	*9*	*0*
AQ230005	0.202122	0.226122	4	15	5	4	13	6	4	12	5

Figure 10.9 presents the precision and recall graph for the full-text query with space as a character. Only the accurate results are visualised here. The actual accuracy value is contained within these charts at the point the fourth and last highly relevant document, i.e. n=4, has been retrieved. Since only the hit threshold parameter has been varied and the same settings were used as in the preliminary results, the average error accuracy does not appear here.

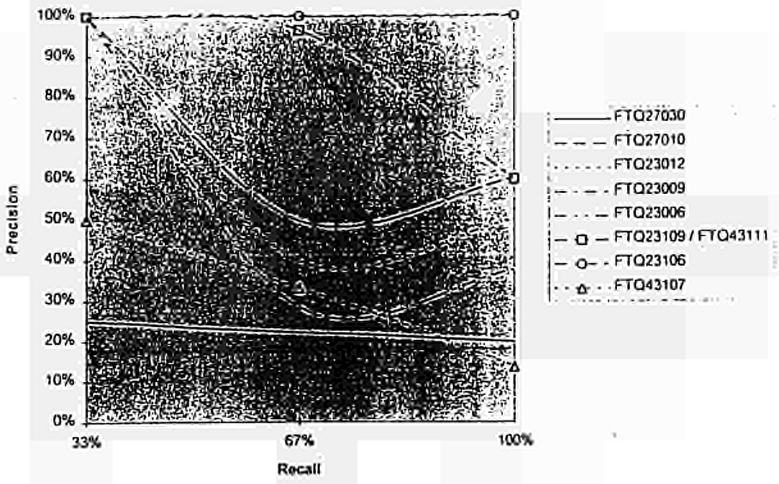


FIGURE 10.9: PRECISION VERSUS RECALL (FULL-TEXT QUERY, SPACE AS A CHARACTER, SORTED ON PERFECT HIT RATE) - ADDITIONAL RESULTS:

To validate the assumption that the hit threshold should be set lower than the maximum map activity, one extremely accurate configuration[42] with a hit threshold lower than the maximum map activity, was re-evaluated with a value, higher than the maximum map activity. This resulted in highly inaccurate outcomes.

Then, the effective range had to be investigated. Therefore the same setting was re-evaluated again, now with a hit threshold, set at more than 50 percent below the minimum map error. The results were extremely accurate.

Next, this experiment was repeated in more detail with one of the highly inaccurate results where a low context size had been used[43]. Accuracy increased as the hit threshold was set lower. When a hit threshold was used of 50 percent below the minimum map error, the state of extreme accuracy was reached.

At this point the experiment was transferred to the only accurate average error-sorted result[44]. Not only did the hit threshold-modifications react identically, but using a hit threshold of 50 percent below the minimum map error, a **perfect** hitlist was retrieved here. Not only were the

[42] Settings properties: full-text query, low generalization factor, high context size, no space as character.

[43] Settings properties: full-text query, low generalization factor, low context size, no space as character.

[44] Settings properties: full-text query, low generalization factor, low context size, space as character.

four most important documents retrieved first, but the next six documents were also significantly correlated.

Until then, the generalisation factor had been kept constant at a low value. Knowing more about the hit threshold, the influence of the generalisation factor was examined in relation to the hit threshold[45]. When hit threshold was used of 50 percent below the maximum map activity, the results were not good at all. However, when a hit threshold of 25 percent was used, the results became extremely accurate.

This left only the artificial query to be optimised, because accurate results with the artificial query would make a comparison with index-based information retrieval systems easier. First an experiment with a configuration with no generalisation factor[46] was carried out in detail. Again, when a hit threshold of 50 percent below the maximum map activity was used, the results became accurate. The hit threshold was then set even lower, until a hit threshold of 88 percent below the maximum map activity. The state of accuracy was continued.

Finally, this experiment was transferred to one of the original artificial settings[47]. Again, when a hit threshold of 50 percent below the maximum map activity was used, the results became accurate. However, when a value of 75 percent below the maximum map activity was used, the results became highly inaccurate again.

Additional Analysis

The hit threshold is the most essential parameter. In general, a value of 50 percent below the maximum map activity gives accurate results. However, as the generalisation factor increases, which causes a decrease in the accuracy of the map's query representation, the hit threshold-value should also increase.

With the context size, robustness and execution time increase. But, *all* preliminary configurations should be optimisable by merely adjusting the hit threshold-value.

The addition of the space as a character increases the information value of the patterns and thus the retrieval quality.

[45] Settings properties: full-text query, medium generalization factor, low context size, space as character.

[46] Settings properties: artificial query, no generalization factor, low context size.

[47] Settings properties: artificial query, low generalization factor, low context size.

The artificial query performs best when it is not compressed.

Comparison

Although the data set was also evaluated with ZyIMAGE, a contemporary index-based information retrieval system, it is not straightforward how a fair comparison can be made with FILTER.

First of all, the data set was composed with ZyIMAGE. This makes a comparison by definition unfair. Also, the number of documents in the data set is too small to compare the systems adequately. A second problem for a fair comparison is the semantic nature of the query's subject. All words starting with the string COMPUT seem to belong to the semantic class of computer terms. Index-based information retrieval systems perform best when these kind of subjects are to be retrieved, because they search for more or less exact matches. The neural filter's performance is subject-insensitive, because it does not search for exact matches, but calculates a correlation.

Also, the best neural filter results were obtained with the full-text query. If the artificial query had yielded the best results, a more valuable comparison could have been made by using exactly the same query in both systems.

Having emphasised the relative importance of any comparison made between these systems, the results, obtained with ZyIMAGE, are reviewed and a comparison is made with the results, obtained with FILTER.

A few Boolean queries were evaluated in the same way as the neural filter queries were evaluated. The approach, taken to retrieve the most accurate results, was to extend the original elementary query by including a few more keywords with the operators AND & OR in such a way that this Boolean query would still represent the semantical core of the full-text query. Perfect retrieval was achieved when the original query was extended with one or two keywords of the set {LINE, DESKTOP, DELL}.

Next, the artificial query itself was evaluated as a quorum query. First without the keyword repetitions, making it an unweighted quorum query with a varying quorum value. The results were highly accurate, until the query became too informative. Next, some the most frequent keyword repetitions were translated into a weighted quorum query. Also Boolean-quorum mixtures were evaluated. In all cases an increase of the information value of the query resulted in a decrease of accuracy.

In all cases, the fuzzy searches returned none or too few documents with these Boolean queries.

In this comparison, with this data set and this query, both in the neural filter and the traditional information retrieval system using Boolean queries, perfect retrieval could be achieved. The artificial query seemed to contain too much information for the index-based system. It only worked well when the artificial query was more or less reduced to a big disjunction of keywords. This high information density was also a problem in the neural filtering environment, but here this problem could be solved quite easily by fine-tuning one or two parameters. In other words, although results were identical in this comparison, the neural filter seems more flexible and more robust.

The results of the comparison experiments are listed in detail in Table 10.9 a,b,c below.

TABLE 10.9 A: COMPARISON WITH ZYIMAGE - ABBREVIATIONS

- F: Fuzzy degree used
- P: Maximum Precision document count
- R: Maximum Recall document count
- E: Error document count at maximum recall

TABLE 10.9 B: ZYIMAGE BOOLEAN QUERIES RESULTS TABLE

Query	F	P	R	E	Comment
COMPUT*	0-4	8	8	0	data composition query
COMPUT* OR DELL	0	6	6	0	
	1-4	8	8	0	
COMPUT* OR DELL*	0-4	6	6	0	
(COMPUT* OR DELL) AND (NOT KEYB*)	0-4	6	-	-	1 essential doc was not found
(COMPUT* OR DELL) AND (NOT SOFTW*)	0	5	5	0	
	1-4	6	6	0	
(COMPUT* OR DELL*) AND (LINE OR DESKTOP)	0	4	4	0	
	1-4	-	-	-	search generated no hits
(COMPUT* OR DELL) AND (LINE OR DESKTOP)		4	4	0	query was not the best hit
	1-4	-	-	-	search generated no hits
COMPUT* AND (LINE OR DESKTOP OR DELL*)	0	4	4	0	
	1-4	2	-	-	only 2 docs were retrieved
COMPUT* AND (LINE OR DESKTOP OR DELL)	0	4	4	0	query was not the best hit
	1-4	-	-	-	search generated no hits

TABLE 10.9 C: ZYIMAGE QUORUM QUERIES RESULTS TABLE

Query	F	P	R	E
1 of {Dell, Desktop, Computers, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier, configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	5	5	0
1 of (...}	1-4	1	-	-
2 of {...}	0	5	5	0
2 of {...}	1-4	-	-	-
3 of {...}	0	5	5	0

4 of {...}	0	5	5	0
5 of {...}	0	6	6	0
6 of {...}	0	1	-	-
2 of {Dell*, Computer*, Desktop} AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	4	-	-
1 of {Dell*, Computer*, Desktop} AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	5	5	0
Dell* AND Computer* AND Desktop AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	2	-	-
Dell* AND Computer* AND 1 of {Desktop, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	2	-	-
Computer* AND 2 of { Dell, Desktop, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	6	6	0

10.4 Discussion

Two important consequences can be drawn from the evaluation:

- The neural filter yields highly accurate results when the parameters are set properly. In the prototype, parameters can be calculated automatically by the hints mechanism. Since this process does not require any additional fine-tuning, it only takes minimum preparation time.

- By using FILTER's table map, maximum execution speed possible can be maintained without compromising retrieval accuracy, which is essential in current awareness applications[48].

Although the neural filter is likely to exhibit more flexible and more robust behaviour than a index-based information retrieval system with respect to a query, as has been pointed out in the comparison section, this primarily holds in a static query environment. In such an environment, a query functions as a user profile. All incoming data passes a series of profiles, all text parts are extracted accordingly and the results are stored in databases.

However, a problem arises when new queries are added. In that case, the system will have to process the whole corpus to obtain an initial update, which can take quite some time, because the incoming data is not made quickly accessible by some sort of indexing. This is the major drawback of the neural filter-algorithm. All data of a text part is needed to determine its correlation with respect to a query, instead of merely locating occurrences of query-strings in the generated index, as is, roughly speaking, the case in index-based retrieval systems.

The optimum corpus preparation for the neural filter system would probably be to store a compressed vector representation of each text part which still contains all data patterns to maintain maximum performance and minimise data storage overhead. By adding the number of occurrences in the text part to each data pattern, all pattern repetitions can be eliminated. This way, the character-to-vector translations and all iterative cycles are eliminated from the corpus extraction process. Although these measures seriously affect the flexibility of the

[48] The data set, used in the evaluation, was processed in about 8 minutes on a 66 Megahertz-486DX2 PC. This means that the FILTER Prototype processes 6 Mb per hour, while maintaining accurate retrieval.

system, because it implies fixed pattern coding parameters, this does not necessarily have to be a problem, because generally applicable parameter-settings have been established in the evaluation.

The neural filter could also be added to existing current awareness applications in a data fusion environment to improve retrieval quality. The idea behind data fusion is that any combination of methods will yield better results than any method applied stand-alone, because each method examines its input from a different perspective, which results in a different output.

In the neural filter, retrieval consists of calculating the correlation of all data patterns in relation to a rigid query representation. In an index-based filter, retrieval consists of locating the query-components in a rigid data index. In other words, the task is approached from quite opposite perspectives. By combining the results of such a compound analysis, more accurate and more robust results are likely to be obtained.

This report has shown that the neural filter can contribute significantly to the class of real-time filtering applications as a high quality full-text search method, **especially in a data fusion environment.**

Part 4

General Discussion & Conclusions

11 General Discussion

Introduction

This chapter discusses the results from the State-of-the-Art Report, the Workshops and the developed prototypes. Although much of the results have already been discussed elsewhere in this report, it is the intention of the writers to provide a short overview of the achieved results. Therefore, this will be a kind of "meta" overview whereas the other discussions were more oriented on details and technical issues.

11.1 State-of-the-Art Report

More than 300 studies towards the application of Artificial Neural Networks (ANN's) in Information Retrieval were investigated in the context of the State-of-the-Art Report. Only a very limited number showed real improvements with respect to existing techniques. The areas in which an improvement with respect to traditional techniques was reported were:

- Fuzzy search on text generated by an Optical Character Recognition (OCR) system [Costello, 1992][Bordogna et al., 1992][Nordell, 1991],
- User modelling in a Selective Dissemination of Information (SDI) context [Scholtes, 1993] [Gallant et al., 1992].
- Searching in multi-medial information [Cawkell, 1993].

Most of the other research projects showed interesting results on "toy problems", but none of the studies showed any real improvements over existing IR technology on large databases as they are used in a library. ANN's had shown appealing properties in the area of clustering and browsing interfaces, but the seriousness of this approach was held to be questionable. The main reason for this were doubts about the scalability of the approach and lack of empirical evaluation.

In order to investigate the probability of success for the application of ANN's in IR, the following three areas where chosen for prototype development:

- Fuzzy searching for OCR
- User Modelling for Filtering in Dynamic Data-Flows
- Bibliographical Clustering for Document Retrieval and Browsing

One could expect improvements of the results by using ANN's in the first two application. However, previous research had not shown much improvements in the application of ANN's for the clustering of bibliographical record or the derivation of a document browser. Nevertheless, since the later is a typical library application, it was expected to provide a real insight in the capabilities of ANN's for typical library problems.

11.2 Prototypes

Neural Networks for Fuzzy Search

Three different methods for fuzzy matching were tested:

- A confusion matrix-,
- A neural network- and
- A wild card search.

Four to seven letter sized, high-frequent words, were used as test set. In total, over 30 megabytes of text generated by different OCR engines was analysed.

In general, the addition of a fuzzy search to an information retrieval engine rarely resulted in the retrieval of documents that were not found without the fuzzy search. main reason for this behaviour is that documents on certain topics always contain more than one occurrence of a typical search word. However, if one needs to know where a word occurs in a document or one involves a relevance ranking algorithm, fuzzy retrieval is absolutely necessary.

For words smaller than four characters, none of the methods seemed to work properly. For words longer than seven characters, the wild card search appeared to be very successful. In order to compare the different techniques, the recall for both the statistical and the neural method was matched to that of the wild card search with fuzzy degree. In this context, recall and precision were measured with respect to the number of words retrieved, NOT to the number of documents retrieved.

For words of length four to six, the statistical as well as the ANN worked much better than the wild card search. In general, the ANN was a little better for words of length four and five. So, it could be stated that ANN's do improve the quality of fuzzy searching. Nevertheless, this comes at a price. In contrary to the wild card search, ANN's need to be trained, they do not find any semantically related words (plurals) and they are slower. Therefore, one should consider whether ANN fuzzy retrieval is worth the trouble.

Outside of the neural network field there are long traditions of research that have produced statistical methods with very good performance. Why cluster with neural networks when there are specialised and reliable cluster algorithms? Why use neural networks for visualisation when there are many other good visualisation tools?

An important advantage of neural networks is their ease of application on underspecified and vaguely defined tasks. They can be used as off-the-shelf methods when the data are not well understood or if data analysis is too expensive. It is however very unlikely that libraries could afford to use them in this way for the construction of vital systems such as classification schemes, retrieval engines or user interfaces.

But, this does not mean that neural networks are useless. We have seen that there are some very useful features of neural networks in the visualisation and clustering domain. Unlike traditional cluster algorithms neural networks fit themselves easily to a combination of cluster based retrieval and visualisation of collection structure. Unlike hierarchical cluster algorithms neural networks do not require the computationally expensive computation of a document similarity matrix. The gradual adjustment of the weight vectors seems to free them of some problems that traditional heuristic cluster algorithms experience. The evaluation results suggest that networks, or network-like cluster algorithms, might be a fair trade-off between the computationally unattractive and the very ineffective extremes of cluster algorithms.

The fact that neural networks can be brought into the league of serious clustering algorithms is encouraging, but it also makes life tough. They have to beat a lot of competitors who are playing the game on their own home ground. We think it is very likely that statistical methods in clustering and browsing will become more 'neural', i.e. incorporate the good ideas from neural networks, in the near future, while at the same time neural networks will become less neural, i.e. incorporate many (statistical) tricks. This can already be seen in the move from the Kohonen map, which was originally inspired by a model of the visual cortex of the brain, to the Fritzke network, which can hardly be called neurobiologically plausible at all.

By testing the Fritzke growing cell network on an artificial and a number of real world data sets, it was found to solve a number of problems that have been identified in the research on clustering with Kohonen maps. We have discussed a number of approaches towards data representation for IR with neural networks, which remain to be investigated more thoroughly. The Gridnet network failed to live up to the demands that IR places on clustering.

The best Fritzke network from our experiments outperformed many well-known cluster algorithms. Thorough experimental evaluation on realistic data sets is the only way neural networks can prove their utility. The test results we have obtained indicate that scalability nor the quality of the clusters are the main problem. The most important limitation is more likely to be the reliability of convergence. Otherwise, neural networks form an attractive alternative to hierarchic clustering algorithms, both from a computational perspective and from the point of view of IR effectiveness. Their usability for visualisation gives neural networks an extra edge over other algorithms which are not suited for this use.

Neural networks do offer some fresh ideas about clustering, but nonetheless it is advisable not to view them as a completely new technology. It is better to treat them equally with their classical counterparts. We expect that statistical methods and neural networks for clustering will continue to enrich each other in the near future, both by sharing valuable techniques and insights, and by setting high quality standards for each other. The proper question for libraries in a setting as described here should not be: "Should we use neural networks?", but rather: "Should we make use of clustering, and if so, what method will do the best job?" A final point to be noted is the fact, however, that in our experiments no clustering algorithm, including the best neural network, could outperform the simple vector space model in retrieval effectiveness.

Filtering Dynamic Data Flows

The data set, with which the Filtering Prototype was evaluated, is a small subset of a corpus of recently scanned newspaper articles. A full-text query and an artificial query were taught to the neural net. The settings were configurations with varying generalisation factors, context sizes, spaces as characters and, although only after the preliminary results, hit thresholds. Three levels of query correlation were determined manually for all data documents.

To determine the correlation between the query representation in the feature map and each document in the data set, a negative filter, a positive filter and a positive-negative filter have been evaluated.

Expected was that the artificial query would yield better results than the full-text query, that an increase of the generalisation factor would decrease accuracy and that the context size and the inclusion of the space as a character would increase accuracy.

In the preliminary results the artificial query didn't do well at all. The full-text query did much better. When the space as a character was used, the results were best. The perfect hit rate turned out to be the most promising correlation calculation, in combination with a high context size. Then the results became extremely accurate. Therefore, the hit threshold was examined in detail in the additional results. Also the artificial query was evaluated again, but this time with no generalisation factor.

In the additional results the hit threshold turned out to be the most important parameter of all. All results could be optimised, so that no irrelevant documents were returned anymore. As a rule of thumb, its value should be set at 50 percent below the maximum map activity. A perfect retrieval was accomplished with the full-text query, where the generalisation factor and the context size were set low, a space as a character was used, the hit threshold was set at 50 percent below the maximum map activity and the hit list was sorted on the perfect hit rate.

After emphasising the relative importance of any comparison made between a neural and a index-based retrieval system, a comparison was made nevertheless. In this comparison, with this data set and this query, also in the index-based information retrieval system using Boolean queries, perfect retrieval was achieved. The artificial query seemed to contain too much information for the index-based system. In general, although these results were identical, the neural filter seems more flexible and more robust.

Although the results were quite promising, some side remarks should be made. This model processes only 6 Mb an hour. It is a non-indexed method, resulting in the need to match all data segments to a ANN per interest model. That is, every interest model is only capable to process 6 Mb an hour. In comparison, the commercial package ZyFILTER runs over 200 Mb a day for several hundreds of queries at a very acceptable level. Therefore, one should question the practical application of this kind of ANN's in a commercial environment.

11.3 Neural Networks for Information Retrieval in a Libraries Context

In all of these applications, extensive comparisons showed that ANN's can hardly do better on "library data" than traditional approaches. In addition, none of the workshop participants could provide suggestions for improvements of the prototypes that would directly lead to outperforming of the traditional models. The only suggestions done were hints for future research.

The main reasons for this failure of ANN's in performance were:

- A too large dimensionality of the typical library data sets. As a result the ANN's did not always reliably converge to proper end-states. One should be aware of the fact that success of ANN's in toy problems does not guarantee success on larger data sets.
- The size of typical library sets is too large to be processed by ANN's. It just takes too long.
- A typical ANN is able to process noisy "natural" signals such as sound or vision very well. Bibliographical records do not contain such data, therefore typical library applications do not use the well-known advantages of ANN's. Their application is often an overkill.

In general, the information stored in libraries does not contain any data that is suited to be processed by ANN's or that could take particular advantages of the ANN's.

In addition, one could state that an ANN is always the second best solution to a problem. That is, an ANN is a good solution if one does not understand the prob..m in all details. However, by the time one does, there is always a better solution. In almost all cases the "better solution" showed to be less - or even non-neural compared to the original ANN model. As stated in the discussion of the cluster prototype, libraries cannot usually compromise for second best systems.

Nevertheless, it might be worthwhile to apply ANN's to certain tasks in a more general "information engineering" setting, because sometime one just doesn't understand the problems that well, or one doesn't have the time to understand the problem.

11.4 Recommendations for Future Research

Although ANN's do not seem to be capable to process the large volumes of information that exist in libraries, and they have not been shown to outperform classical (statistical pattern recognition and information retrieval) methods by a significant margin, we still see some interesting areas for further investigation. These areas are mostly badly understood domains, in which no major efforts to employ traditional techniques have been made. Some possible extensions of traditional information retrieval models could be:

- Extend the relevance ranking schemes with more in-depth and context sensitive knowledge and make these algorithms more aware of vague and undefined relations that can be learned from examples. Here, document vectors obtained from a traditional system can be trained to e.g. self-organising neural networks.
- Combine multiple relevance ranking schemes with a so-called data fusion model. Here, pairs of user relevance feedback on the quality of retrieved documents with respect to their relevance ranking values can be trained to a neural network. This very non-linear mapping will adopt various relevance ranking schemes to the specific preferences of the end-user.
- Data mining bibliographical record databases in order to clean up doubles and "out-of-context" classification assignments.

In all these cases, the ANN can be used as a device that is capable to learn a small mapping. That is, the ANN is able to increase the relevance ranking algorithm by taking small user preferences in account, the ANN can combine several relevance ranking algorithms in a very non-linear algorithm in order to adopt even more to the specific user model, and the ANN is capable to learn typical record patterns. However, these are niche "information engineering" solutions, that focus more on the technical Information Retrieval problems than on the typical library problems.

Extending Relevance Ranking

Here the purpose of the prototype is to extend the retrieval quality by increasing the ranking algorithm and making it more context sensitive, less noise sensitive and giving it more generalisation capabilities. This could be done by training all vectors, representing bibliographical records and full-text documents, to a self-organising feature map, and use the formed structures as additional information in the relevance ranking phase. In most cases, the

SOFM will derive a structure where related documents are within the same area of the map. If some document is retrieved, either the system or the user could provide neighbouring documents as "relevant" alternatives. One can either provide these documents automatically, but can could also "browse" through the network, jumping from one relevancy cluster to another.

Data Fusion

There is no perfect relevance ranking algorithm. Every relevance ranking algorithm measures another property of the record or document. Even the above described additional neural ranking will sometimes be wrong. So why not combine the different relevance ranking schemes such as term-based frequency ranking (mainly measuring term distribution), the vector space model VSM (mainly measuring important words), and a neural ranking into one super ranking by using data fusion. However, relevance ranking quality is a very personal measure, is different for every person and the combination of different relevance ranking values into one super-value is a very non-linear mapping.

So why use an extended set of document relevance ranking values and train them to a supervised neural net, thereby making an internal mapping of the specific user (off course there can always be a common denominator user).

Input:	(R1, R2, R3, ..., Rn)	normalised relevance ranking values for several algorithms
Output	**R**	super relevance value
Train:	{(R1, R2, R3, ..., Rn), **R**}	pairs of relevance ranking values with user defined **R**'s

Research presented in [Bartell, 1994] has shown that quality improvements of at least 40% can be obtained.

Data Mining Bibliographical Records

In addition, ANN's can be used to clean up the database. This is, especially bibliographical databases contain several errors in misspellings of names, institutes, periodicals, etc. Often, the same objects are assigned different entities.

Insurance companies, credit card companies, banks, large mailing houses and other institutions maintaining extended databases are already familiar with ANN's implementing data mining technology in order to clean up their databases and trace down errors.

The great benefit of ANN's in this context is that they are able to discover "wrong" or "inappropriate-for-context" classification information.

11.5 Guidelines for the Application of Neural Networks in Information Retrieval in a Libraries Context

The most important conclusion of the present study is that Artificial Neural Networks are have not been found to represent an *enabling technology* in any domain of IR that we have reviewed. I.e. ANN's do not constitute solutions to problems that were hitherto considered unsolvable. At the most, ANN's can be an *enabling metaphor*, i.e they can provide an original new perspective on problems whose solution has seemed hard to define in traditional computational terms (e.g. semantic road maps, associative thesauri, interest profile storage etcetera).

Moreover, the similarities of ANN's with various traditional techniques, especially statistical pattern recognition methods, are much more abundant than their differences. Often it can even be considered a matter of personal taste whether a particular approach can be called a neural network approach. Once a problem has been casted in terms of a neural network, it can usually be solved either more efficiently or more effectively by some related, but less "neural" and more specialised method. Therefore, if one is concerned with the performance of a system in a demanding real world environment, the decision whether to use ANN's for particular problems in Information Retrieval cannot be based on a superficial choice between "paradigms". The advantages of one (neural) method over another usually lie in the finer technical details.

This does by no means implicate that neural networks cannot or should not be considered as one of many suitable candidate techniques for solving pattern recognition and learning tasks in IR.

But, when applying ANN's, one should be aware to use them for applications that take advantage of typical ANN properties. In this volume we have come across a number of important guidelines. Here we suffice with summarising the most important ones.

- Artificial Neural Networks can automatically derive complex non-linear mappings. The price paid for this powerful behaviour is unreliability of convergence. However, in most cases the tasks in IR can just as well be handled by linear mappings.

- ANN's handle badly-defined problems by training from collected examples. This behaviour is, however, not uncommon among statistical pattern recognition techniques of various sorts. The performance of ANN's has to be evaluated empirically, using traditional evaluation methodology, against a background provided by statistical techniques.

- Keep the dimensionality of the problem small (sometimes this can be done by data compression), as ANN training is very time consuming and can become unstable in large scale application domains

- Work on as much "natural" data as possible.

- ANN's work best on noisy data sets, whereas traditional methods often have higher requirements for the quality of the data sets.

- ANN's are often suitable for problems where one has a limited time-frame in order to solve the problem. If the data cannot be thoroughly analysed, the non-parametric properties of ANN's can be of good use.

Most of the problems faced in present day libraries do not belong to this class. And, as of now it is uncertain whether problems with the above mentioned properties will become part of librarianship. However, as libaries are faced with such rapid changes nowadays, this posibility cannot yet be excluded.

Amsterdam,, March 28th, 1995

Dr Johannes C. Scholtes, M.S.C. Information Retrieval Technologies BV

Bibliography

[Allen, 1990]: Allen, R.B. (1990). Connectionist Language Users. Connection Science, Vol. 2, No. 4, pp. 279-312.

[Allen, 1991]: Allen, R.B. (1991). Knowledge Representation and Information Retrieval with Simple Recurrent Networks. Working Notes of the AAAI SSS on Connectionist Natural Language Processing. March 26-28, Palo Alto, CA., pp. 1-6.

[Anderson, 1972]: Anderson, J.A. (1972). A Simple Neural Network Generating an Interactive Memory. Mathematical Biosciences, Vol. 14, pp. 197-220.

[Anderson, 1983]: Anderson, J.A. (1983). Cognitive and Psychological Computation with Neural Models. IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, pp. 799-815.

[Anthes, 1993]: Anthes, G.H. (1993). How the Feds Find Data: Retrieval System More Intuitive than Traditional Search Tools. Computerworld. Vol. 27, No. 35. pp. 47-49.

[Aristotle, ca. 400 BC]: Aristotle (ca. 400 BC). "De Memoria et Reminiscentia", Aristotle on Memory, Richard Sorabji (Translation). Brown University Press.

[Bahrami et al., 1992]: Bahrami, A. and Dagli, C.H. (1992). Design retrieval by fuzzy neurocomputing. Journal of Engineering Design. Vol. 3, No. 4, pp. 339-356.

[Baldi et al., 1989]: Baldi, P. and Hornik, K. 1989, “Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima.”, Neural Networks, Vol. 2, p.53-58.

[BANKAI, 1991]: Intelligent Information Access. Proceedings of the BANKAI Workshop. 14-16 Oct. 1991. Brussels, Belgium.

[Barthes et al., 1991]: Barthes, C., Cohen, P., Glize, P., and Machonin, A. (1991). Neural computation in knowledge based systems. Digital Signal Processing - 91. Proceedings of the International Conference. pp. 509-513. 4-6 Sept. 1991. Florence, Italy.

[Bein et al., 1988]: Bein, J. and Smolensky, P. (1988). Applications of the Interactive Activation Model to Document Retrieval. Technical Report CU-CS-405-88, University of Colorado, Boulder, CO.

[Belew, 1986]: Belew, R.K. (1986). Adaptive Information Retrieval: Machine Learning in Associative Networks. Ph.D. Thesis, Univ. Michigan, CS Department, Ann Arbor, MI.

[Belew, 1987]: Belew, R.K. (1987). A Connectionist Approach to Conceptual Information Retrieval. Proceedings of the First International Conference on Artificial Intelligence and Law, pp. 116-126. ACM Press.

[Belew, 1989]: Belew, R.K. (1989). Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn About Documents. Proceedings of the 12th ACM-SIGIR Conference on Research & Development in Information Retrieval. June 11-20. Cambridge, MA, pp. 11-20.

[Belew et al., 1988]: Belew, R.K. and Holland, M.P. (1988). A Computer System Designed to Support the Near-Library User of Information Retrieval. Microcomputers for Information Management, Vol. 5, No. 3, pp. 147-167.

[Bellcore, 1991]: Bellcore Workshop on High Performance Information Filtering. November 5-7, Chester, NJ.

[Bernstein, 1981]: Bernstein, J. (1981). Profiles: AI, Marvin Minsky. The New Yorker, December 14, pp. 50-126.

[Bichsel et al., 1989]: Bichsel, M. and Seitz, P. (1989). Minimum Class Entropy: A Maximum Information Approach to Layered Networks. Neural Networks, Vol. 2, No. 2, pp. 133-141.

[Biennier et al., 1990]: Biennier, F., Guivarch, M., and Pinon, J.-M. (1990). Browsing in hyperdocuments with the assistance of a neural network. Proceedings of the First European Conference on Hypertext. pp. 288-297. 27-30 Nov. 1990. Versailles, France.

[Biennier et al., 1990]: Biennier, F., Pinon, J.-M., and Guivarch, M. (1990). A connectionist system to assist navigation in hyperdocuments. Neuro-Nimes '90. Third International Workshop. Neural Networks and Their Applications. pp. 539-550. 12-16 Nov. 1990. Nimes, France.

[Biennier et al., 1990]: Biennier, F., Pinon, J.M., and Guivarch, M. (1990). A connectionist method to retrieve information in hyperdocuments. INNC 90 Paris. International Neural Network Conference. Vol. 1, pp. 444-448. 9-13 July 1990. Paris, France.

[Biennier et al., 1992]: Biennier, F. and Favrel, J. (1992). Dynamic knowledge systems for new production trends. IFIP Transactions B (Applications in Technology). Vol. B-3, pp. 301-310. 24-26 June 1992. Tokyo, Japan.

[Blackmore et al., 1993]: Blackmore, Justine and Risto Miikkulainen, 1993, "Incremental Grid Growing: Encoding High Dimensional Structure into a Two-Dimensional Feature Map", Proc. of the 1993 IEEE International Conference on Neural Networks, San Francisco, CA, p.450-455.

[Blair et al., 1985]: Blair, D.C. and Maron, M.E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. Communications of the ACM, Vol. 28, No. 3, pp. 289-299.

[Blanchard, 1992]: Blanchard, D. (1992). Informix and Excalibur Sign Joint Agreement. AI Expert. Vol. 7, No. 7. July 1992.

[Bochereau Laurent et al., 1991]: Bochereau Laurent (1991). Extracting Legal Knowledge by Means of a Multi Layer Neural Net. Proceeding of the International Conference on Artificial Intelligence in Law,

[Bokhari, 1981]: Bokhari, S.H. (1981). On the Mapping Problem. IEEE Transactions on Computers, Vol. 30, No. 3, pp. 207-214.

[Bordogna et al., 1992]: Bordogna, G. and Pasi, G. (1992). A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and its Evaluation. Journal of the American Society of Information Science. Vol. 44, No. 2, pp. 70-82.

[Bounds, 1989]: Bounds, D. (1989). Expert Systems and Connectionist Networks. In: Connectionism in Perspective (R. Pfeiffer et al., Eds.), pp. 277-282. North-Holland.

[Boyer et al., 1977]: Boyer, R.S., Moore, J.S. (1977). A Fast String Searching Algorithm. Communications of the ACM, Vol. 20, No. 10, pp. 762-772.

[Bozinovic et al., 1982]: Bozinovic, R., Srihari, S.N. (1982). A String Correction Algorithm for Cursive Script Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov. 1982, pp. 655-663, pp. 236-244.

[Brachman et al., 1988]: Brachman, R.J. and McGuinness, D.L. (1988). Knowledge Representation, Connectionism, and Conceptual Retrieval. Proceedings of the 11th ACM-SIGIR Conference on Research & Development in Information Retrieval, pp. 161-174.

[Bradshaw et al., 1989]: Bradshaw, G., Fozzard, R. and Ceci, L. (1989). A Connectionist Expert System That Actually Works. In: Advances in Neural Information Processing Systems (D.S. Touretzky, Editor), Vol. 1, pp. 248-255. Morgan Kaufmann.

[Bryson et al., 1969/1975]: Bryson, A.E. and Ho, Y-C. (1969). Applied Optimal Control. Hemisphere Publishing.

[Busch, 1992]: Busch, E. (1992). Search and Retrieval. How to Evaluate Large Text-Retrieval Systems. Byte. June 1992. pp. 271-276.

[Buta, 1994]: Buta, P. (1994). Mining for Financial Knowledge with CBR. AI Expert. Vol. 9, No. 2, pp. 34-41.

[Carlson, 1991]: Carlson, P.A. (1991). Virtual text and new habits of mind. New Results and New Trends in Computer Science. pp. 25-53. 20-21 June 1991. Graz, Austria.

[Carpenter at al., 1987a]: Carpenter, G.A. and S. Grossberg (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing. Vol. 37. Academic Press. pp. 54-115.

[Carpenter at al., 1987b]: Carpenter, G.A. and S. Grossberg (1987). ART2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics. Vol. 26. Optical Society of America. pp. 4919-4930.

[Carpenter et al., 1988]: Carpenter, G.A. and Stephen Grossberg, 1988, “The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network”, IEEE Computer, March, p.77-88.

[Carpenter at al., 1990]: Carpenter, G.A. and S. Grossberg (1990). ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Networks. Vol. 3. pp. 129-152.

[Carpenter at al., 1991a]: Carpenter, G.A., S. Grossberg, and D.B. Rosen (1991). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. Neural Networks. Vol. 4. pp. 759-771.

[Carpenter at al., 1991b]: Carpenter, G.A., S. Grossberg, and J.R. Reynolds. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Networks. Vol. 4. pp. 565-588.

[Carpenter at al., 1991c]: Carpenter, G.A., and S. Grossberg (eds.) (1991). Pattern recognition by self-organizing neural networks. MIT Press.

[Caudell, 1992]: Caudell, T.P. (1992). Genetic algorithms as a tool for the analysis of adaptive resonance theory network training sets. COGANN-92. International Workshop on

Combinations of Genetic Algorithms and Neural Networks (Cat. No.92TH0435-8). pp. 184-200. 6 June 1992. Baltimore, MD.

[Caudell, 1992]: Caudell, T.P. (1992). Hybrid opto-electronic adaptive resonance theory neural processor, ART1. Applied Optics. Vol. 31, No. 29, pp. 6220-6229.

[Caudell et al., 1991]: Caudell, T.P., Smith, S.D.G., Escobedo, R., and Johnson, G.C. (1991). A neural database system for reusable engineering. 1991 IEEE International Joint Conference on Neural Networks (Cat. No.91CH3065-0). Vol. 1, pp. 254-261. 18-21 Nov. 1991, Singapore.

[Cawkell, 1993]: Cawkell, A.E. (1993). The British Library's Picture Research Projects. Advanced Imaging 1993. pp. 38-39.

[Ceccarelli et al., 1992]: Ceccarelli, M.; Petrosino, A. (1992). Information retrieval in sparse associative memories. Artificial Neural Networks, 2. Proceedings of the 1992 International Conference (ICANN-92). Vol. 1, pp. 297-301.

[Charniak, 1983]: Charniak, E. (1983). Passing Markers: A Theory of Contextual Influence in Language Comprehension. Cognitive Science, Vol. 7, pp. 171-190.

[Chen, 1992]: Chen, Q. and Norcio, A.F. (1992). Modelling users with neural architectures. IJCNN International Joint Conference on Neural Networks (Cat. No.92CH3114-6). Vol. 1, pp. 547-552. 7-11 June 1992. Baltimore, MD.

[Chen et al., 1993]: Chen, H., Lynch, K.J., Basu, K., and Ngo, T.D. (1993). Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval. IEEE Expert, April 1993, pp. 25-34.

[Chen et al., 1992]: Chen, O.T.-C., Sheu, B.J., Fang, W.C. (1992). Information Processing & Management. Vol. 28, No. 6, pp. 687-706.

[Cherkassky et al., 1992]: Cherkassky, V., Vassilas, N., Brodt, G.L., and Wechsler, H. (1992). Conventional and associative memory approaches to automatic spelling correction. Engineering Applications of Artificial Intelligence. Vol. 5, No. 3, pp. 223-237.

[Cherkassky et al., 1990]: Cherkassky, V., Vassilas, N., and Brodt, G.L. (1990). Conventional and associative memory-based spelling checkers. Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence (Cat. No.90CH2915-7). pp. 138-144. 6-9 Nov. 1990. Herndon, VA.

[Chrisley, 1990]: Chrisley, R.L. (1990). Cognitive Map Construction and Use: A Parallel Distributed Processing Approach. In: Connectionist Models: Proceedings of the 1990 Summer School (D.S. Touretzky, J.L. Elman, T.J. Sejnowski and G.E. Hinton, Eds.), pp. 287-300. Morgan Kaufmann.

[Cleeremans et al., 1989]: Cleeremans, A., Servan-Schreiber, D. and McClelland, J.L. (1989). Finite State Automata and Simple Recurrent Networks. Neural Computation, Vol. 1, No. 3, pp. 372-381.

[Cleeremans et al., 1990]: Cleeremans, A., McClelland, J.L. (1990). Learning the Structure of Event Sequences. Proceedings of the 12th Annual Conference of the Cognitive Science Society, July 25-28, Cambridge, MA, pp. 709-716.

[Cochet et al., 1988]: Cochet, Y. and Paget, G. (1988). ZZENN: A New Approach to Connectionist Machine Learning. Proceedings of the International Computer Science Conference, Artificial Intelligence: Theory and Applications, Hong Kong, pp. 377-380.

[Cohen et al., 1987]: Cohen, P.R. and Kjeldsen, R. (1987). Information Retrieval By Constrained Spreading Activation in Semantic Networks. Information Processing & Management, pp. 255-268.

[Computer Letters, 1993]: Computer letters (1993). Getting Good Help: Software Agents, No Matter How They Are Defined, Will Be needed to Get the Most Mileage out of the Coming Information Highway. Computer letter, Vol. 9, No. 29. pp. 1-7.

[Cooper, 1971]: Cooper, W.S. (1971). A Definition of Relevance for Information Retrieval. Information Storage and Retrieval, Vol. 7, No. 1, pp. 19-37.

[Costello, 1992]: Costello, J. (1992). That Fuzzy Way of Thinking. Computer Weekly. June 25, 1992. pp. 28.

[Cottrell, 1989]: Cottrell, G.W. (1989). A Connectionist Approach to Word Sense Disambiguation. Morgan Kaufmann.

[Cottrell et al., 1987]: Cottrell, G.W., P. Munro and D. Zipser, (1987), Learning Internal Representations from Grey-Scale Images: an Example of Extensional Programming. Ninth Annual Conference of the Cognitive Science Society, Seattle, WA, pp. 462-473.

[Crestani, 1993]: Crestani, F. (1993). An adaptive information retrieval system based on neural networks. New Trends in Neural Computation. International Workshop on Artificial Neural Networks. IWANN '93 Proceedings. pp. 732-737.

[Croft, 1977]: Croft, W.B. (1977). Clustering Large Files of Documents Using the Single Link Method. Journal of the American Society for Information Science (JASIS), Vol. 28, No. 6, pp. 341-344.

[Croft, 1980]: Croft, W.B. (1980). A Model of Cluster Searching Based on Classification. Information Systems, Vol. 5, No. 3, pp. 189-195.

[Croft, 1981]: Croft, W.B. (1981). Document Representation in Probabilistic Models in Clustered Files. Journal of the American Society for Information Science (JASIS), Vol. 31, No. 6, pp. 451-457.

[Croft et al., 1979]: Croft, W.B., Harper, D.J. (1979). Using Probabilistic Models of Information Retrieval without Relevance Information. Journal of Documentation, Vol. 35, No. 4, pp. 285-295.

[Croft et al., 1991]: Croft, W.B., Turtle, H.R. and Lewis, D.D. (1991). The Use of Structured Queries in Information Retrieval. Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval. October 13-16, Chicago, Illinois. pp. 32-45.

[Crough, 1986]: Crough, D.B. (1986). The Visual Display of Information in an Information Retrieval Environment. Proceedings of the 9th International ACM -SIGIR Conference on Research and Development in Information Retrieval, pp. 58-67.

[D'Amore et al., 1988]: D'Amore, R. and Mah, C. (1988). N-Grams. PAR Government Systems Corp., McLean, Virginia.

[DARPA, 1991]: DARPA, Defense Advanced Research Project Agency (1991). Message Understanding Conference (MUC-3). Proceedings Of the Third Message Understanding Conference (MUC-3), DARPA.

[Deffner et al., 1990]: Deffner, R. and Geiger, H. (1990). Associative word recognition with Neural Networks. Tools for Knowledge Organisation and the human interface. Proceedings of the 1st International ISKO Conference, Darmstadt, 14-17 Aug, Vol. 1, pp. 80-87.

[Deffner et al., 1990]: Deffner, R. and Geiger, H. (1990). Associative word recognition with connectionist architectures. Proceedings. 1st International ISKO-Conference. pp. 80-87. 14-17 Aug. 1990. Darmstadt, West Germany.

[Deffner et al., 1991]: Deffner, R. and Geiger, H. (1991). Neural nets understand natural language. 5. Language processing-a challenge for AI research. Elektronik. Vol. 40, No. 5, pp. 106-113.

[Desrocques et al., 1991]: Desrocques, G., Bassano, J.-C., and Archambault, D. (1991). An associative neural expert system for information retrieval. Intelligent Text and Image Handling. RIAO '91. pp. 546-566. 2-5 April 1991. Barcelona, Spain.

[Devijver et al., 1982]: Devijver, P.A. and Kittler, J. (1982). Pattern Recognition, A Statistical Approach. Prentice Hall.

[Dolan et al., 1987]: Dolan, C.P. and Dyer, M.G. (1987). Symbolic Schemata, Role Binding, and the Evolution of Structure in Connectionist Memories. Proceedings of the IEEE International Neural Network Conference. June 21-24. San Diego, CA, pp. 287-298.

[Dontas et al., 1990]: Dontas, K., Sarma, J., Srinivasan, P., and Wechsler, H. (1990). Fault tolerant hashing and information retrieval using back propagation. Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences. Vol. 4, pp. 345-352. 2-5 Jan. 1990, Kailua-Kona, HI.

[Dorffner, 1988]: Dorffner, G. (1988). NETZSPRECH - Another case for Distributed 'Rule' Systems. Proceedings of the 10th Annual Conference of the Cognitive Science Society, August 17-19. Montreal, Canada, pp. 573-579.

[Dorffner, 1991]: Dorffner, G. (1991). "Radical" Connectionism for Natural Language Processing. Working Notes of the AAAI Spring Symposium Series, Palo Alto, CA, pp. 95-106.

[Doszkocs, 1992]: Doszkocs, T.E. (1992). Neural Networks in Libraries. Library and Information Technology Association, 3rd National Conference on Information Technology, Denver, CO, pp. 81-83.

[Doszkocs, 1991]: Doszkocs, T.E. (1991). Library applications of neural networks. Computers in Libraries '91. Proceedings of the 6th Annual Computers in Libraries Conference. pp. 44-48.

[Doszkocs et al., 1990]: Doszkocs, T.E., Reggia, J. and Lin, X. (1990). Connectionist Models and Information Retrieval. Annual Review of Information Science and Technology, Vol. 25, pp. 209-260.

[Doyle, 1961]: Doyle, L.B. (1961). Semantic Road Maps for Literature Searchers. Journal of the ACM, Vol. 8, pp. 553-578.

[Duda et al., 1973]: Duda, R.O. and Hart, P.E. (1973). Pattern Classification and Scene Analysis. John Wiley & Sons.

[Dumais et al., 1988]: Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwater, S. and Harshman, R. (1988). Using Latent Semantic Analysis to Improve Access to Textual Information. ACM, CHI'88, pp. 281-285.

[Edelman, 1987]: Edelman, G.M. (1987). Neural Darwinism: The Theory ofNeuronal Group Selection. Basic Books.

[Eichmann et al., 1991]: Eichmann, D.A. and Srinivas, K. (1991). Neural Network-Based Retrieval from Reuse Repositories. CHI '91 Workshop on Pattern Recognition and Neural Networks in Human-Computer Interaction, April 28. New Orleans, LA.

[Eichmann et al., 1992]: Eichmann, D.A. and Srinivas, K. (1992). Neural Network-Based Retrieval from Reuse Repositories. In: Neural Networks and Pattern Recognition in Human Computer Interaction (R. Beale and J. Findlay, Eds.), Ellis Horwood.

[Eliot, 1992]: Eliot, L.B. (1992). Don't forget the hardware. AI Expert. Vol. 7, No. 9. September 1992.

[Eliot, 1993]: Eliot, L.B. (1993). Tuning a Database Expertly. AI Expert. Vol. 8, No. 1, pp. 9-11.

[Elman, 1988]: Elman, J.L. (1988). Finding Structure in Time. Cognitive Science, Vol. 14, No. 2, pp. 179-212.

[Elman, 1991a]: Elman, J.L. (1991). Distributed Representations, Simple Recurrent Networks and Grammatical Structure. Machine Learning. Special Issue on Connectionist Approaches to Language Learning, Vol. 7, No. 2/3, pp. 195-226.

[Elman, 1991b]: Elman, J.L. (1991). Incremental Learning, or The Importance of Starting Small. Proceedings of the 13th Annual Conference of the Cognitive Science Society, August 7-10, Chicago, IL, pp. 443-448.

[Emigh, 1992]: Emigh, Jacqueline (1992). New for the PC: Windows Version of Info Select Shipping. Newsbytes November 2, 1992.

[EUROMICRO, 1992]: Eighteenth EUROMICRO Symposium on Microprocessing and Microprogramming (EUROMICRO 92). Microprocessing & Microprogramming. Vol. 35, No. 1-5. Eighteenth EUROMICRO Symposium on Microprocessing and Microprogramming (EUROMICRO 92). 14-17 Sept. 1992. Paris, France.

[Feldman, 1981]: Feldman, J.A. (1981). A Connectionist Model of Visual Memory. In: Parallel Models of Associative Memory (G.E. Hinton and J.A. Anderson, Eds.), pp. 65-97. Lawrence Erlbaum.

[Feldman, 1989]: Feldman, J.A. (1989). What Lies Ahead. Byte, January 1989, pp. 348-350.

[Feldman et al., 1982]: Feldman, J.A. and Ballard, D.H. (1982). Connectionist Models and their Properties. Cognitive Science, Vol. 6, pp. 205-254.

[Fodor et al., 1988]: Fodor, J.A. and Pylyshyn, Z.W. (1988). Connectionism and Cognitive Architecture: A Critical Analysis. In: Connections and Symbols (J.A. Fodor and Z.W. Pylyshyn, Eds.), MIT Press.

[Ford, 1989]: Ford, N. (1989). From Information - to Knowledge Management; the Role of Rule Induction and Neural Network Machine Learning Techniques in Knowledge Generation. Journal of Information Science, Vol. 15, No. 4/5, pp. 299-304.

[Forney, 1973]: Forney Jr., G.D. (1973). The Viterbi Algorithm. Proceedings of the IEEE, Vol. LXI, pp. 268-278.

[Foster, 1992]: Goster, G. (1992). Expert systems to help the help desk. Business Communications Review. Vol. 22, No. 10, pp. 40-43.

[Fritzke, 1991a]: Fritzke, B. (1991). Let it Grow - Self-Organizing Feature Maps with Problem Dependent Cell Structure. In: Artificial Neural Networks (T. Kohonen, K. Makisara, O. Simula and J. Kangas, Eds.), Vol. 1, pp. 403-408. Elsevier Science Publishers.

[Fritzke, 1992a]: Fritzke, B. (1992). Wachsende Zellstrukturen - Ein Selbstorganisierendes Neuronales Netzwerkmodell (In German). Ph.D. Thesis, Universitat Erlangen-Nurnberg.

[Fritzke, 1992b]: Fritzke, B. (1992). Growing Cell Structures - a Self-organizing Network in k Dimensions. In: Artificial Neural Networks 2 (I. Aleksander and J. Taylor, Eds.), Vol. 2, pp. 1051-1056. Elsevier Science Publishers.

[Fritzke et al., 1991]: Fritzke, B. and Wilke, P. (1991). FLEXMAP - A Neural Network For The Travelling Salesman Problem With Linear Time and Space Complexity. Proceedings of the International Joint Conference on Neural Networks, November 18-21, Singapore, pp. 929-934.

[Fritzke, 1993]: Fritzke Bernd, 1993, "Growing Cell Structures - A Self-Organizing Network for Unsupervised and Supervised Learning", International Computer Science Institute, Berkeley, CA, TR-93-026.

[Fu, 1977]: Fu, K.S. (Editor) (1977). Syntactic Pattern Recognition, Applications. Springer-Verlag.

[Gacem et al., 1990]: Gacem, A., Maren, A., and Uhrig, R. (1990). A neural network to extract implicit knowledge from a nuclear data base. Transactions of the American Nuclear Society. Vol. 62, pp. 129. 11-15 Nov. 1990. Washington, DC.

[Gallant, 1988]: Gallant, S.I. (1988). Connectionist Expert Systems. Communications of the ACM, Vol. 31, No. 2, pp. 152-169.

[Gallant et al., 1992]: Gallant, S.I., Caid, W.R., Carleton, J., Hecht-Nielsen, R., Kent Pu Qing, and Sudbeck, D. (1992): HNC's Match Plus system. Text Retrieval Conference (TREC-1) (NIST-SP-500-207). pp. 107-111. 4-6 Nov. 1992. Gaithersburg, MD.

[Gallant et al., 1992]: Gallant, S.I., Caid, W.R., Carleton, J., Hecht-Nielsen, R., Pu Qing, K., and Sudbeck, D. (1992) HNC's MatchPlus system (document retrieval). SIGIR Forum Vol. 26, No. 2, pp. 34-38.

[Gedeon et al., 1991]: Gedeon, T.D. and Mital, V. (1991). Information retrieval in law using a neural network integrated with hypertext. 1991 IEEE International Joint Conference on Neural Networks (Cat. No.91CH3065-0). Vol. 2, pp. 1819-1824. 18-21 Nov. 1991, Singapore.

[Germain, 1992]: Germain, E. (1992). Introducing Natural Language Processing (Tutorial). AI Expert. Vol. 7, No. 8. August 1992.

[Gersho et al., 1990a]: Gersho, M. and Reiter, R. (1990). Information Retrieval using Self-Organizing and Heteroassociative Supervised Neural Network. Proceedings of the International Neural Network Conference, July 9-13, Paris, France. Vol. 1, pp. 361-364.

[Gersho et al., 1990b]: Gersho, M. and Reiter, R. (1990). Information Retrieval using a Hybrid Multi-Layer Neural Network. Proceedings of the International Joint Conference on Neural Networks, June 17-21, San Diego, CA. Vol. 2, pp. 111-117.

[Gigley, 1983]: Gigley, H.M. (1983). HOPE -- AI and the Dynamic Process of Language Behaviour. Cognition and Brain Theory, Vol. 6, No. 1.

[Gigley, 1985]: Gigley, H.M. (1985). Computational Neurolinguistics-What is it all about? Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp. 260-266.

[Gilliland, 1993]: Gilliland, S. (1993). Info Select 1.0. Windows Sources. Vol. 1, No. 5. pp. 264-265.

[Ginsberg, 1993]: Ginsberg, A. (1993). A Unified Approach to Automatic Indexing and Information Retrieval. IEEE Expert. Vol. 8, No. 5. pp. 46-57.

[Goldberg, 1989]: Goldberg, D.E. (1989). Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley Publishing Company.

[Goldberg et al., 1988]: Goldberg, D.E. and Holland, J.H. (Editors) (1988). Machine Learning (Special Issue on Genetic Algorithms). Vol. 3, No. 2-3.

[Griffiths et al., 1986] : Griffiths, A.H., Luckhurst, C., and Willett, P. (1986). Using Interdocument Similarity Information in Document Retrieval Systems. Journal of the American Society for Information Science. Vol. 37, No. 1, pp. 3-11.

[Grossberg, 1976a]: Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. Biological Cybernetics. Vol. 23. Springer Verlag. pp. 121-134.

[Grossberg, 1976b]: Grossberg, S. (1976). Adaptive pattern classification and universal recoding. II: Feedback. expectation. olfaction. illusions. Biological Cybernetics. Vol. 23. Springer Verlag. pp. 187-202.

[Gutknecht et al., 1990]: Gutknecht, M. and Pfeifer, R. (1990). An Approach to Integrating Expert Systems with Connectionist Networks. AI Communications, Vol. 3, No. 3, pp. 116-127.

[Hanson et al., 1974]: Hanson, A.R., Riseman, E.M. (1974). A Contextual Post-processing System for Error Correction Using Binary n-Grams. IEEE Transactions on Computers, May 1974, pp. 480-493, pp. 141-154.

[Harman, 1993]: Harman, Donna, 1993, "Overview of the first Text REtrieval Conference", Proc. of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA, p.36-48.

[Harrison, 1971]: Harrison, M.C. (1971). Implementation of the Substring Test by Hashing. Communications of the ACM, Vol. 14, No. 12, pp. 777-779.

[Hauser et al., 1993]: Hauser, S.E., Hsu, W., and Thoma, G.R. (1993). Request routing with a back error propagation network. Applications of Artificial Neural Networks IV. Proceedings of the SPIE - The International Society for Optical Engineering. Vol. 1965, pp. 689-94. 13-16 April 1993, Orlando, FL.

[Hayes-Roth, 1985]: Hayes-Roth, B. (1985). A Blackboard System for Control. Artificial Intelligence, Vol. 26, pp. 251-321.

[Hebb, 1949]: Hebb, D.O. (1949). The Organization of Behavior: A Neuropsychological Theory. John Wiley & Sons.

[Hecht-Nielsen, 1990]: Hecht-Nielsen, R. (1990). Neurocomputing. Addison Wesley Publishing Company.

[De Heer, 1982]: De Heer, T., 1982, "The Application of the Concept of Homeosemy to Natural Language Information Retrieval", Information Processing and Management, Vol. 18(5), p.229-236.

[Heger et al., 1991]: Heger, A.S. and Koen, B.V. (1991). Good relationships are pivotal in nuclear data bases. Nuclear Safety. Vol. 32, No. 4, pp. 488-493.

[Hemmje et al., 1994]: Hemmje, Matthias, Clemens Kunkel and Alexander Willett, 1994, "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval", Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, p.249-259.

[Hendler, 1989]: Hendler, J.A. (1989). Spreading Activation over Distributed Microfeatures. In: Advances in Neural Information Processing Systems (D.S. Touretsky, Editor), Vol. 1, pp. 553-559. Morgan Kaufmann.

[Henseler, 1993]: Henseler, J. (1993). Neurons, Connection and Activations. Ph.D. Thesis, University of Maastricht, The Netherlands.

[Hingston et al., 1990]: Hingston, P. and Wilkinson, R. (1990). Document Retrieval Using a Neural Network. Technical Report, Department of Computer Science, RMIT at the University of Melbourne, 1990.

[Hingston et al., 1990]: Hingston, P. and Wilkinson, R. (1990). Document retrieval using a neural network. AI '90. Proceedings of the 4th Australian Joint Conference on Artificial Intelligence. pp. 304-310. 21-23 Nov. 1990. Perth, WA, Australia.

[Hinton, 1981]: Hinton, G.E. (1981). Implementing Semantic Networks in Parallel Hardware. In: Parallel Models of Associative Memory, (G.E. Hinton and J.A. Anderson, Eds.), Lawrence Erlbaum.

[Hinton et al., 1989]: Hinton, G.E. and Anderson, J.A. (Eds.) (1989). Parallel Models of Associative Memory, Updated Edition. Lawrence Erlbaum.

[Hopfield, 1982]: Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Science, Vol. 79, pp. 2554-2558.

[Hull et al., 1982]: Hull, J.J., Srihari, S.N. (1982). Experiments in Text Recognition with Binary n-Grams and Viterbi Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 520-530.

[Hurt, 1991]: Hurt, C.D. (1991). Fuzzy cognitive mapping of online search strategies. 12th National Online Meeting. Proceedings 1991. pp. 157-162. 7-9 May 1991. New York, NY.

[Husek et al., 1992]: Husek, D. and Pokorny, J. (1992). Spreading activation methods in information retrieval-a connectionist approach. Neurocomputing. Vol. 4, No. 1-2, pp. 31-36.

[Iwadera et al., 1992]: Iwadera, T. and Kimoto, H. (1992). The effects of a dynamic word network on information retrieval. Proceedings of the SPIE - The International Society for Optical Engineering. Vol. 2, pp. 622-630. 21-24 April 1992, Orlando, FL.

[Jagota, 1990]: Jagota, A. (1990). Applying a Hopfield-Style Network to Degraded Text Recognition. Proceedings of the International Joint Conference on Neural Networks 1990, Vol. 2, pp. 607-610.

[Jennings et al., 1992]: Jennings, A.and Higuchi, H. (1992). A Browser with a Neural Network User Model. Library Hi-Tech, Vol. 10, No. 1-2, pp. 77-93.

[Jennings et al., 1993]: Jennings, A., Higuchi, H., and Liu, H. (1993). A user model neural network for a personal news service. Australian Telecommunication Research. Vol. 27, No. 1, pp. 1-12.

[Jennings et al., 1993]: Jennings, A. and Higuchi, H. (1993). A user model neural network for a personal news service. User Modeling and User-Adapted Interaction, Vol. 3, No. 1, pp. 1-25.

[Jennings et al., 1992]: Jennings, A. and Higuchi, H. (1992). A personal news service based on a user model neural network. IEICE Transactions on Information and Systems. Vol. E75-D, No. 2, pp. 198-209.

[Johnston et al., 1991]: Johnston, M. and Weckert, J. (1991). Machine learning for library monograph selection. Expert Systems for Information Management. Vol. 4, No. 2 pp. 77-91

[Johnston et al., 1990]: Johnston, M. and Weckert, J. (1990). Expert assistance for collection development. Libraries and Expert Systems. Proceedings of a Conference and Workshop. pp. 70-87.

[Jordan, 1986a]: Jordan, M.I. (1986). Serial Order: A Parallel Distributed Processing Approach. Institute for Cognitive Science, Technical Report #8604, UCSD, La Jolla, CA.

[Jordan, 1986b]: Jordan, M.I. (1986). Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. Proceedings of the Cognitive Science Society, Amherst, MA, pp. 531-546.

[Jung et al., 1991]: Jung, G.S. and Raghavan, V.V. (1991). Connectionist Learning in Constructing Thesaurus-Like Knowledge Structure. AAAI Symposium on Text-Based Inteligent Systems, Stanford University, Palo Alto, CA.

[Kacprzyk, 1992]: Kacprzyk, J. (1992). Fuzzy logic with linguistic quantifiers in decision making and control. Archives of Control Sciences. Vol. 1 (37), No. 1-2, pp. 127-141.

[Kamimura, 1990a]: Kamimura, R. (1990). Experimental Analysis of Performance of Temporal Supervised Learning Algorithm, Applied to a Long and Complex Sequence. Proceedings of the International Neural Network Conference, July 9-13, Paris, France, pp. 753-756.

[Kamimura, 1990b]: Kamimura, R. (1990). Application of Temporal Supervised Learning Algoritm to Generation of Natural Language. in: Proceedings of the IEEE International Joint Conference on Neural Networks, June 17-21, San Diego, CA. Vol. 1, pp. 201-207.

[Kangas, 1992]: Kangas, J.A. (1992). Temporal Knowledge in Locations of Activations in a Self-Organizing Map. In: Artificial Neural Networks 2 (I. Aleksander and J. Taylor, Eds.), Vol. 1, pp. 117-120. Elsevier Science Publishers.

[Kantor, 1993]: Kantor, P.B. (1993). The adaptive network library interface: a historical view and interim report. Library Hi Tech, Vol. 11, No. 3, pp. 81-92.

[Kelly, 1991]: Kelly, D.T. (1991). Developments in Automated Recognition and its Commercial Implications. Information Media and Technology, Vol. 24, No. 6, Nov 1991, pp. 256-260.

[Kelly, 1991]: Kelly, M. (1991), Self-organising map training using dynamic k-d trees, Proceedings of the ICANN '91, June 24th-28th, Helsinki, pp. 1041-1044.

[Kimbrell, 1988]: Kimbrell, R.E. (1988). Searching for Text? Send an N-Gram! Byte, May 1988, pp. 297-312.

[Kimoto, 1993]: Kimoto, H. and Iwadera, T.I. (1993). Associated information retrieval system (AIRS)-its performance and user experience. IEICE Transactions on Information and Systems vol.E76-D, No. 2, pp. 274-83.

[Kimoto et al, 1991]: Kimoto, H. and Iwadera, T. (1991). A dynamic thesaurus and its application to associated information retrieval. IJCNN-91-Seattle: International Joint Conference on Neural Networks (Cat. No.91CH3049-4). Vol. 1, pp. 19-29. 8-14 July 1991. Seattle, WA.

[Knuth et al., 1977]: Knuth, D.E., Morris, J.H., Pratt, V.R. (1977). Fast Pattern Matching In Strings. SIAM Journal of Computing, Vol. 6, No. 2, pp. 323-350.

[Koch, 1992]: Koch, C. (1992). Combining connectionist and hypertext techniques in the study of texts: a HyperNet approach to literary scholarship. Literary & Linguistic Computing. Vol. 7, No. 4, pp. 209-217.

[Kodratoff et al., 1990]: Kodratoff, Y. and Michalski, R.S. (Eds.) (1990). Machine Learning, An Artificial Intelligence Approach. Volume 3. Morgan Kaufmann.

[Koekebakker, 1991]: Koekebakker, O. (1991). SMR Stelt Inzicht in Brein Neurale Netwerken Voorop (in Dutch). Computable. October 11, 1991. pp 29.

[Kohonen, 1972]: Kohonen, T. (1972). Correlation Matrix Memories. IEEE Transactions on Computers, Vol. C-21, pp. 353-359.

[Kohonen, 1977]: Kohonen, T. (1977). Associative Memory: A Systems-Theoretical Approach. Springer-Verlag.

[Kohonen, 1982a]: Kohonen, T. (1982). Analysis of a Simple Self-Organizing Process. Biological Cybernetics, Vol. 44, pp. 135-140.

[Kohonen, 1982b]: Kohonen, T. (1982). Clustering, Taxonomy, and Topological Maps of Patterns. Proceedings of the 6th International Conference on Pattern Recognition, pp. 114-128.

[Kohonen, 1982c]: Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics, Vol. 43, pp. 59-69.

[Kohonen, 1984]: Kohonen, T. (1984). Self-Organization and Associative Memory. Springer-Verlag.

[Kohonen, 1988]: Kohonen, T. (1988). An Introduction to Neural Computing. Neural Networks, Vol. 1, nr. 1, pp. 3-16.

[Kohonen, 1990a]: Kohonen, T. (1990). Some Practical Aspects of the Self-Organizing Maps. Proceedings of the International Joint Conference on Neural Networks, January 15-19, Washington DC, Vol. 2, pp. 253-256.

[Kohonen, 1990b]: Kohonen, T. (1990). The Self-Organizing Map. Proceedings of the IEEE, Vol. 78, pp. 1464-1480.

[Kohonen, 1991]: Kohonen, T. (1991). The Hypermap Architecture. In: Artificial Neural Networks (T. Kohonen, K. Makisara, O. Simula and J. Kangas, Eds.), Vol. 2, pp. 1357-1362. Elsevier Science Publishers.

[Koikkalainen, 1990]: Koikkalainen, P., Oja, E. (1990), Self-organising hierarchical feature maps, Proceedings of the IJCNN, San Diego, June 17-21, 1990, pp. 279-284.

[Kosko, 1992]: Kosko, B. (1992). Neural Networks and Fuzzy systems: a Dynamical Systems Approach to Machine Intelligence. Prentice-Hall Inc.

[Kuhnel et al., 1990]: Kuhnel, H., Tavan, T. (1990). The Anti-Hebb Rule Derived from Information Theory. In: Parallel Processing in Neural Systems and Computers (R. Eckmiller, G. Hartmann and G. Hauske, Eds), pp. 187-190. Elsevier Science Publishers.

[Kukich, 1988]: Kukich, K. (1988). Back Propagation Topologies for Sequence Generation. Proceedings of the IEEE International Conference on Neural Networks, San Diego, Vol. 1, pp. 301-308.

[Kwok, 1989]: Kwok, K.L. (1989). A Neural Network for Probabilistic Information Retrieval. Proceedings of the 12th ACM-SIGIR Conference on Research & Development in Information Retrieval. June 11-20, 1989. Cambridge, MA, pp. 21-30.

[Kwok, 1990]: Kwok, K.L. (1990). Application of Neural Network to Information Retrieval. Proceedings of the International Joint Conference on Neural Networks, January 15-19. Washington DC. Vol. 2, pp. 623-626.

[Kwok, 1991a]: Kwok, K.L. (1991). Query Modification and expanding in a Network with Adaptive Architecture. Proceedings of the 14th Annual International ACM/SIGIR Conference on Research & Development in Information Retrieval, Chicago, IL, pp. 192-201.

[Kwok, 1991b]: Kwok, K.L. (1991). Query learning using an ANN with adaptive architecture. Machine Learning. Proceedings of the Eighth International Workshop on Machine Learning (ML91). pp. 260-264. June 199. Evanston, IL.

[Lakoff, 1988]: Lakoff, G. (1988). A Suggestion for a Linguists with Connectionist Foundations. In: Proceedings of the Connectionist Models Summer School (D.S. Touretsky, G.E. Hinton and T.J. Sejnowski, Eds.), Carnegie Mellon University, PA, pp. 301-314. Morgan Kaufmann.

[Lancaster, 1979]: Lancaster, F.W. (1979). Information Retrieval Systems: Characteristics, Testing and Evaluation (Second Edition). John Wiley & Sons.

[Larson, 1988]: Larson, P.A. (1988). Dynamic Hash Tables. Communications of the ACM, Vol. 31, nr. 4, pp. 446-457.

[Le Cun, 1986]: Le Cun, Y. (1986). Learning Process in an Asymmetric Threshold Network. In: Disordered Systems and Biological Organizations (E.Bienenstock, F. Fogelman Souli, and G. Weisbuch, Eds.), Springer-Verlag.

[Lee et al., 1993]: Lee, D.L. and Croft, W.B. (1993). Artificial Intelligence in Text-Based Information Systems. IEEE Expert. April 1993, pp. 6-7.

[Lelu, 1991] : Lelu, A. (1991). From Data Analysis to Neural Networks: New Prospects for Efficient Browsing through Databases. Journal of Information Science, Vol. 17, pp. 1-12.

[Lelu, et al., 1992]: Lelu, A. and Francois, C. (1992). Information retrieval based on a neural unsupervised extraction of thematic fuzzy clusters. Fifth International Conference. Neural Networks and their Applications. NEURO NIMES 92. pp. 93-104. 2-6 Nov. 1992. Nimes, France.

[Levine, 1992]: Levine, R. (1992). The State of Diagnostics. MIDRANGE Systems, Vol. 5, No. 13. July 7, 1992.

[Lewinson, 1994]: Lewinson, L. (1994). Data Mining: Tapping into the Mother Lode. Database Programming & Design. Vol. 7, No. 2, pp. 50-51.

[Liddy et al., 1991]: Liddy, E.D. and Paik, W. (1991). Automatic recognition of semantic relations in text. Structuring of Information. Informatics 11. pp. 65-77.

[Lim et al., 1992]: Lim, E.-P. and Cherkassky, V. (1992). Semantic networks and associative databases: two approaches to knowledge representation and reasoning. IEEE Expert. Vol. 7, no. 4, pp. 31-40.

[Lin et al., 1991]: Lin, X., Soergel, D., and Marchionini, G. (1991). A Self-organizing Semantic Map for Information Retrieval. Proceedings of the 14th Annual International ACM-SIGIR Conference on Research & Development in Information Retrieval, Chicago, IL. pp. 262-269.

[Linsker, 1987]: Linsker, R. (1987). Towards an Organizing Principle for a Layered Perceptual Network. In: Neural Information Processing Systems (D.Z. Anderson, Editor), pp. 485-494. Amererican Institute of Physics.

[Linsker, 1988]: Linsker, R. (1988). Self-Organization in a Perceptual Network. IEEE Computer, Special Issue on Artificial Neural Systems, Vol. 21, nr. 3, pp. 105-117.

[Linsker, 1989a]: Linsker, R. (1989). An Application of the Principle of Maximum Information Preservation to Linear Systems. In: Advances in Neural Information Processing Systems (D.S. Touretzky, Editor), Vol. 1, pp. 186-194. Morgan Kaufmann.

[Linsker, 1989b]: Linsker, R. (1989). How to Generate Ordered Maps by Maximizing the Mutual Information between Input and Output Signals. Neural Computation, Vol. 1, pp. 396-405.

[Linsker, 1990]: Linsker, R. (1990). Perceptual Neural Organization: Some Approaches Based on Network Models and Information Theory. Annual Review of Neuroscience, Vol. 13, pp. 257-281.

[Lippmann, 1987]: Lippmann, R.P. (1987). An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, April 1987, pp. 4-22.

[MacLeod, 1990]: MacLeod, K. (1990). Neural Architectures for Clustering in Document Databases. PhD Dissertation. Technical University of Nova Scotia. Halifax, Nova Scotia, Canada.

[MacLeod, 1990]: MacLeod, K. (1990). An application specific neural model for document clustering. Proceedings of the Fourth Annual Parallel Processing Symposium. Vol. 1, pp. 5-16. 4-6 April 1990. Fullerton, CA.

[MacLeod et al., 1991]: Macleod, K.J. and Robertson, W. (1991). A Neural Algorithm for Document Clustering. Information Processing & Management. Vol. 27, No. 4, pp. 337-346.

[Malsberg, 1973]: Malsberg, Chr. von der (1973). Self-Organization of Orientation Sensitive Cells in the Striate Cortex. Kybernetik, Vol. 14, pp. 85-100.

[Marcus, 1980]: Marcus, M.P. (1980). A Theory of Syntactic Recognition for Natural Language. MIT Press.

[Mauldin, 1991]: Mauldin, M.L. (1991). Conceptual Information Retrieval: A Case Study in Adaptive Partial Parsing. Kluwer Academic Publishers.

[McClelland et al., 1981]: McClelland, J.L. and Rumelhart, D.E. (1981). An Interactive Activation Model of Context Effects in Letter Perception. Psychological Review, Vol. 88, pp. 375-407.

[McCormick, 1992]: McCormick, J. (1992). Excalibur Gets First Big Contract. Newsbytes, May 7, 1992.

[McCulloch et al., 1943]: McCulloch, W.S. and Pitts, W.A. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematics and Biophysics, Vol. 5, pp. 115-133.

[McIllroy, 1982]: McIllroy, M.D. (1982). Development of a Spelling List. IEEE Transactions on Communications, Vol. COM-30, pp. 91-99.

[Michalski et al., 1986a]: Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.) (1986). Machine Learning, An Artificial Intelligence Approach. Volume 1. Morgan Kaufmann.

[Michalski et al., 1986b]: Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.) (1986). Machine Learning, An Artificial Intelligence Approach. Volume 2. Morgan Kaufmann.

[Miikkulainen et al., 1988a]: Miikkulainen, R. and Dyer, M.G. (1988). Encoding Input/Output Representations in Connectionist Cognitive Systems. In: Proceedings of the Connectionist Models Summer School (D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski, Eds.), Carnegie Mellon University, PA, pp. 347-356. Morgan Kaufmann.

[Miikkulainen et al., 1988b]: Miikkulainen, R. and Dyer, M.G. (1988). Forming Global Representations with Extended Back Propagation. Proceedings of the IEEE International Conference on Neural Networks, San Diego, CA.

[Miller, 1956]: Miller, A.M. (1956). The Magical Number Seven, Plus or Minus Two: Sone Limits to our Capacity for Processing Information. Psychological Review, Vol. 63, pp. 81-97.

[Minsky et al., 1969/1988]: Minsky, M.L. and Papert, S. (1969). Perceptrons (2nd Edition). MIT Press.

[Mital, 1991]: Mital, V. and Gedeon, T.D. (1991). A neural network integrated with hypertext for legal document assembly. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences (Cat. No.91TH0394-7), Vol. 4, pp. 533-539. 7-10 Jan. 1992. Kauai, HI.

[Mital et al., 199x]: Mital, V. and Gedeon, T.D. (199x). Automating Text Anlysis and Information Retrieval in Law using a Neural Network. Proceedings of the 13th British Computer Society's Information Retrieval Research Coloquium. Section 5.

[Mitzman et al., 1990]: Mitzman, D. and Giovannini, R. (1990). ActivityNets: A Neural Classifier of Natural Language Descriptions of Economic Activities. Proceedings of the International Workshop on Neural Nets for Statistics and Economic Data, December 10-11, 1991. Dublin, Ireland.

[Monthe, 1992]: Mothe, J., (1992). SYRENE: an information retrieval system based on neural approaches: experimental results. Fifth International Conference. Neural Networks and their Applications. NEURO NIMES 92. pp. 81-91. 2-6 Nov. 1992. Nimes, France.

[Moore, 1988]: Moore, B. (1988). ART 1 and Pattern Clustering. Connectionist Models Summer School. Pittsburg, PA.

[Morch, 1992]: Morch, F. (1992). Hvem soger efter dokumentation in ar 2000..."[Who retrieves documentation in the year 2000; the intermediary or the end user?]. Technology and Competence, Proceedings of the 8th Nordic Conference on Information and Documentation, Helsingborg, 19-21 May 1992, Karin Adler, E.Helmer & H.I.Holm (Eds.).

[Mori et al., 1991]: Mori, H., Kinoe, Y., Seto, K., and Hayashi, Y. (1991). Cooperative document retrieval making user's Ill-defined query evolve. International Journal of Human-Computer Interaction. Vol. 3, No. 3, pp. 253-266.

[Mori et al., 1990]: Mori, H., Chung, C.L., Kinoe, Y., and Hayashi, Y. (1990). An adaptive document retrieval system using a neural network. International Journal of Human-Computer Interaction. Vol. 2, No. 3, pp. 267-280.

[Mothe et al., 1992]: Mothe, J.; Soule-Dupuy, C. (1992). Integration of a connectionist model in information retrieval systems. Artificial Neural Networks, 2. Proceedings of the 1992 International Conference (ICANN-92). Vol. 2, pp. 1611-1614.

[Mozer, 1984]: Mozer, M.C. (1984). Inductive Information Retrieval Using Parallel Distributed Computation. Technical Report TR-ICS 8406, UCSD, La Jolla, CA.

[Myers-Tierney, 1992]: Myers-Tierney, L. (1992). An Introduction to Text Management; a Guide for the Office Users. Patricia's Seybold's Office Computing Report. Vol. 14, No. 10. pp. 8-19.

[Neuhoff, 1975]: Neuhoff, D.L. (1975). The Viterbi Algorithm As an Aid in Text Recognition. IEEE Transactions on Information Theory, Vol. IT-XXI, pp. 222-226.

[Newquist, 1994]: Newquist, H.P. (1994). More Notes on Route 666: Navigating the Information Superhighway Will be no Day at the Beach. AI Expert. Vol. 9, No. 3, pp. 41-44.

[Nordell, 1991]: Nordell, D. (1991). Inexact Terminologies: The better to Mimic the Brain (Artificial Neural Networks). Inform. Vol 5, pp. 31-33, April 1991.

[Nowakowska, 1990]: Nowakowska, M. (1990). Cluster analysis, graphs, and branching processes as new methodologies for intelligent systems on example of bibliometric and social network data. International Journal of Intelligent Systems. Vol. 5, No. 3, pp. 247-263.

[Oakes et al., 1993]: Oakes, M.P. and Reid, D. (1993). Some Practical Applications of Neural Networks in Information Retrieval. Proceedings of the 13th British Computer Society's Information Retrieval Research Coloquium. Section 5.

[Oddy et al., 1991]: Oddy, R.N. and Balakrishnan, B. (1991). PThomas: an Adaptive Information Retrieval System on the Connection Machine. Information Processing and Management.

[Olsen, 1994]: Olsen, F. (1994). For Problem Diagnosis, Molly's Help Desk Program Puts on a Tinking Cap. Government Computer News. Vol 13, No. 3, pp. 29.

[Osborn, 1992]: Osborn, T. (1992). Sidebar: Artificial Neural Networks, Library Hi-Tech, Vol. 10, No. 1-2, pp. 70.

[Palakal et al., 1991]: Palakal, M. and Thai, X. (1991). Computational Model of a Biologically Plausible Cognitive map. Proceedings of the International Joint Conference on Neural Networks. July. 8-12, 1991. Seattle, WA. Vol. 2, pp. A-885.

[Parker, 1985]: Parker, D. (1985). Learning Logic. Technical Report TR-87, Center for Computational Research in Economics and Management Science, MIT Press.

[Perez, 1991]: Perez, E.R. (1991). Neural Network Applications for Library Management (Presented at CIL Conference 1991). Library Software Review, Vol. 10, pp. 349-350.

[Personnaz et al., 1986]: Personnaz, L., Guyon, I. and Dreyfus, G. (1986). Neural Network Design for Efficient Information Retrieval. In: Disordered Systems and Biological Organization (E. Bienenstock et al., Eds.), Springer-Verlag.

[Quast, 1992]: Quast, D. (1992). IR - System Baserade pa ny Neurala Netverk och Nunskapsrepresentation"[IR Systems in Lybraries based on Research of Neural Networks and Knowledge Representation.]. Technology and Competence, Proceedings of the 8th Nordic Conference on Information and Documentation, Helsingborg, 19-21 May 1992, Karin Adler, E.Helmer & H.I.Holm (Eds.), pp. 83-87.

[Raan et al., 1993]: Raan, A.F.J.V. and Tijssen, R.J.W. (1993). The Neural Net of Neural Network Research: an Exercise in Bibliometric Mapping. Scientometrics, Vol. 26, No. 10, Jan 1993, pp. 169-192.

[Rapp et al., 1990]: Rapp, R. and Wettler, M. (1990). Simulation of search term generation in information retrieval by propagation in a connectionist lexical net. Nachrichten für Dokumentation. Vol. 41, No. 1, pp. 27-32.

[Rasmussen, 1991]: Rasmussen, E.M. (1991). Introduction: Parallel Processing and Information Retrieval. Information Processing & Management. Vol. 27, No. 4, pp. 255-263.

[Rasmussen, 1992]: Rasmussen, E.M. (1992). Parallel Information Processing. Annual Review of Information Science and Technologt (ARIST). Vol. 27, 1992, pp. 99-130.

[Reeke et al., 1988]: Reeke, G.N. and Edelman, G.M. (1988). Real Brains and Artificial Intelligence. In: The AI Debate. False Starts, Real Foundations (R. Graubard, Editor), pp. 143-174. MIT Press.

[Regier, 1988]: Regier, T. (1988). Recognizing Image-Schemes Using Programmable Networks. In: Proceedings of the Connectionist Models Summer School, (D.S. Touretzky, G.E. Hinton, and T.J. Sejnowski, Eds.), Carnegie Mellon University, PA, pp. 315-324. Morgan Kaufmann.

[Reilly et al., 1990]: Reilly, K.D., Jun Ming Zhan, Mercado, J., and Man-Li Chang (1990). Neural net modelling in diagnosis and information systems. Proceedings of the 1990 Summer Computer Simulation Conference. pp. 677-679. 16-18 July 1990. Calgary, Alta., Canada.

[Rice et al., 1992]: Rice, S.V., Kanai, J. and Nartker, T.A. (1992). A Report on the Accuracy of OCR Devices. Technical Report 92-02. Information Science Research Institute. University of Nevada. Las Vegas, NV.

[Rice et al., 1994]: Rice, S.V., Kanai, J. and Nartker, T.A. (1994). A Report on the Accuracy of OCR Devices. Technical Report 94-02. Information Science Research Institute. University of Nevada. Las Vegas, NV.

[Rijsbergen, 1979]: Rijsbergen, C.J. van (1979). Information Retrieval. Butterworths, London.

[Ritter et al., 1989a]: Ritter, H. and Schulten, K. (1989). Convergency Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability and Dimension Selection. Biological Cybernetics, Vol. 60, pp. 59-71.

[Ritter et al., 1989b]: Ritter, H. and Kohonen, T. (1989). Self-Organizing Semantical Maps. Biological Cybernetics, Vol. 61, pp. 241-254.

[Ritter et al., 1990]: Ritter, H. and Kohonen, T. (1990). Learning "Semantotopic Maps" from Context. Proceedings of the International Joint Conference on Neural Networks, January 15-19, Washington DC. Vol. 1, pp. 23-26.

[Ritter et al., 1992]: Ritter, H., Martinetz, T., Schulten, K. (1992). Neural Computation and Self-organizing Maps, An Introduction. Addison Wesley Publishing Company.

[Robbins et al., 1951]: Robbins, H. and Monro, S. (1951). A Stochastic Appriximation Method. Annals of Math. Stat., Vol. 22, pp. 400-407.

[Robertson et al., 1993]: Robertson, A.M. and Willett, P. (1993). Evaluation of techniques for the conflation of modern and seventeenth century English spelling. Proceedings of the BCS 14th Information Retrieval Colloquium. 13-14 April 1992. Lancaster, UK.

[Rose, 1990]: Rose, D.E. (1990). Appropriate Uses of Hybrid Systems. In: Connectionist Models: Proceedings of the 1990 Summer School (D.S. Touretzky, J.L. Elman, T.J. Sejnowski, G.E. Hinton), pp. 277-286. Morgan Kaufmann.

[Rose, 1991]: Rose, D.E. (1991). A Symbolic and Connectionist Approach to Legal Information Retrieval. Ph.D. Thesis, UCSD, La Jolla, CA.

[Rose et al., 1989]: Rose, D.E. and Belew, R.K. (1989). A case for Symbolic/Sub-Symbolic Hybrids. Proceedings of the 11th Annual Conference of the Cognitive Science Society, August. 16-19, 1989. Ann Arbor, MI. pp. 844-851.

[Rose et al., 1991]: Rose, D.E. and Belew, R.K. (1991). A Connectionist and Symbolic Hybrid for Improving Legal Research. International Journal of Man-Machine Studies.

[Rosenblatt, 1958]: Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, Vol. 65, pp. 386-408.

[Rosenblatt, 1962]: Rosenblatt, F. (1962). Principles of Neurodynamics. Spartan.

[Rumelhart et al., 1982]: Rumelhart, D.E. and McClelland, J.L. (1982). An Interactive Activation Model of Context Effects in Letter Perception: Part 2. Psychological Review, Vol. 89, pp. 60-94.

[Rumelhart et al., 1986a]: Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986). Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing. Vol. 1. (D.E. Rumelhart and J.L. McClelland, Eds.), pp. 318-362. MIT Press.

[Saito et al., 1988]: Saito, K. and Nakano, R. (1988). Medical Diagnostic Expert System Based on PDP Model. Proceedings of the IEEE International Conference on Neural Networks, 1988, San Diego, CA.

[Salton, 1968]: Salton, G. (1968). Automatic Information Organization and Retrieval. McGraw-Hill Book Co., New York.

[Salton, 1971]: Salton, G. (Editor) (1971). The Smart Retrieval System. Prentice Hall.

[Salton, 1972]: Salton, G. (1972). Experiments in Automatic Thesaurus Construction for information Retrieval. Information Processing & Management, Vol. 71, pp. 115-123.

[Salton, 1980a]: Salton, G. (1980). Automatic Term Class Construction Using Relevance - A Summery of Work in Automatic Pseudoclassification. Information Processing & Management. 1980; 16(1), pp. 1-15.

[Salton, 1980b]: Salton, G. (1980). Automatic Information Retrieval. IEEE Computer, Vol. 13, No. 9, pp. 41-57.

[Salton, 1981]: Salton, G. (1981). A Blueprint for Automatic Indexing. ACM SIGIR Forum, 1981 Fall; 16(2), pp. 22-38.

[Salton, 1986]: Salton, G. (1986). Another Look at Automatic Text Retrieval. Communications of the ACM, Vol. 29, No. 7, pp. 648-656.

[Salton, 1989]: Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Publishing Company.

[Salton et al., 1994]: Salton, G., Allan, J., and Buckley, C. (1994). Communications of the ACM. Volume 37, No. 2, pp. 97-109.

[Salton et al., 1968]: Salton, G., Wong, A., Yang, C.S. (1968). A Vector Space Model for Automatic Indexing. Communications of the ACM, Vol. 18, No. 11, pp. 613-620.

[Salton et al., 1973]: Salton, G., Yang, C.S. (1973). On the Specification of Term Values in Automatic Indexing. Journal of Documentation, Vol. 29, no. 4, pp. 351-372.

[Salton et al., 1983a]: Salton, G. and McGill, M.J. (Eds.) (1983). Introduction to Modern Information Retrieval. McGraw-Hill.

[Salton et al., 1983b]: Salton, G., Fox, E.A., Wu, H. (1983). Extended Boolean Information Retrieval. Communications of the ACM, Vol. 26, No. 11, pp. 189-195.

[Salton et al., 1985]: Salton, G., Fox, E.A. and Voorhees, E. (1985). Advanced Feedback Methods in Information Retrieval. Journal of the American Society for Information Science, Vol. 36, No. 3, pp. 200-210.

[Salton et al., 1987]: Salton, G. and Buckley, C. (1987). Term Weighting Approaches in Automatic Text Retrieval. Information Processing & Management, 1987, 24, pp. 513-523.

[Salton et al., 1988a]: Salton, G. and Buckley, C. (1988). On the Use of Spreading Activation Methods in Automatic Information Retrieval. Proceedings of the 11th International ACM-SIGIR Conference on Research & Development in Information Retrieval, June 13-15, Grenoble, France, pp. 147-160.

[Salton et al., 1988b]: Salton, G. and Buckley, C. (1988). Parallel Text Search Methods. Communications of the ACM, Vol. 31, No. 2, pp. 202-215.

[Salton et al., 1991]: Salton, G. and Buckley, C. (1991). Automatic Text Structuring and Retrieval: Experiments in Automatic Encyclopedia Searching. Proceedings of the 14th Annual International ACM-SIGIR Conference on Research & Development in Information Retrieval, Chicago, IL. pp. 21-31.

[Sammon, 1969]: Sammon, J.W. (1969). A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers, Vol. 18, pp. 401-409.

[Savoy, 1992]: Savoy, J. (1992). Bayesian Inference Networks and Spreading Activation in Hypertext Systems. Information Processing & Management. Vol. 28, No. 3, pp. 389-406.

[Scholtes, 1991a]: Scholtes, J.C. (1991). Using Extended Kohonen-Feature Maps in a Language Acquisition Model. Proceedings of the 2nd Australian Conference on Neural Nets. February 2-4. Sydney, Australia, pp. 38-43.

[Scholtes, 1991b]: Scholtes, J.C. (1991). Learning Simple Semantics by Self-Organization. Worknotes of the AAAI Spring Symposium Series on Machine Learning of Natural Language and Ontology. March 26-29. Palo Alto, CA, pp. 146-151.

[Scholtes, 1991c]: Scholtes, J.C. (1991). Learning Simple Semantics by Self-Organization. Worknotes of the AAAI Spring Symposium on Connectionist Natural Language Processing. March 26-29. Palo Alto, CA, pp. 78-83.

[Scholtes, 1991d]: Scholtes, J.C. (1991). Recurrent Kohonen Self-Organization in Natural Language Processing. In: Artificial Neural Networks (T. Kohonen, K. Makisara, O. Simula and J. Kangas, Eds.), pp. 1751-1754. Elsevier Science Publishers.

[Scholtes, 1991e]: Scholtes, J.C. (1991). Unsupervised Context Learning in Natural Language Processing. Proceedings of the International Joint Conference on Neural Networks. July 8-12. Seattle, WA, Vol. 1, pp. 107-112.

[Scholtes, 1991f]: Scholtes, J.C. (1991). Kohonen's Self-Organizing Map in Natural Language Processing. Proceedings of the SNN Symposium. May 1-2. Nijmegen, The Netherlands, pp. 64.

[Scholtes, 1991g]: Scholtes, J.C. (1991). Kohonen's Self-Organizing Map Applied Towards Natural Language Processing. Proceedings of the CUNY 1991 Conference on Sentence Processing. May 12-14. Rochester, NY, pp. 10.

[Scholtes, 1991h]: Scholtes, J.C. (1991). Kohonen Feature Maps in Natural Language Processing. Technical Report ITLI-CL-1, Department of Computational Linguistics. March 1991, University of Amsterdam.

[Scholtes, 1991i]: Scholtes, J.C. (1991). Self-Organized Language Learning. The Annual Conference on Cybernetics: Its Evolution and Its Praxis. July 17-21. Amherst, MA.

[Scholtes, 1993]: Scholtes, J.C. (1993). Neural Networks in Natural Language Processing and Information Retrieval. PhD Thesis. University of Amsterdam, Department of Computational Linguistics, Faculty of Arts, Amsterdam, The Netherlands.

[Schütze, 1993]: Schütze, H., 1993, "Distributed Syntactic Representantions with an Application to Part-of-Speech Tagging", Proc. of the 1993 IEEE International Conference on Neural Networks, San Francisco, CA, Vol.3, p.1504-1509.

[Schwartz, 1993]: Schwartz, K.D. (1993). Why Electronic Filing Systems are Replacing Metal Ones. Government Computer News. Vol. 12, No. 21. pp. 56.

[Sejnowski et al., 1986]: Sejnowski, T.J. and Rosenberg, C.R. (1986). NETtalk: A Parallel Network That Learns to Read Aloud. Technical Report JHU/EECS-86/01, John Hopkins University.

[Servan-Schreiber et al., 1988]: Servan-Schreiber, D., Cleeremans, A. and McClelland, J.L. (1988). Encoding Sequential Structure in Simple Recurrent Networks. Technical Report TR CMU-CS-88-183, Carnegie-Mellon University, PA.

[Servan-Schreiber et al., 1989]: Servan-Schreiber, D., Cleeremans, A. and McClelland, J.L. (1989). Learning Sequential Structure in Simple Recurrent Networks. In: Advances in Neural Information Processing Systems (D.S. Touretsky, Editor), Vol. 1, pp. 643-652. Morgan Kaufmann.

[Servan-Schreiber et al., 1991]: Servan-Schreiber, D., Cleeremans, A. and McClelland, J.L. (1991). Graded State Machines: The Representation of Temporal Contingencies in Simple Recurrent Networks. Machine Learning. Special Issue on Connectionist Approaches to Language Learning. Vol. 7, No. 2-3, pp. 161-194.

[Shannon, 1948]: Shannon, C.E. (1948). A Mathematical Theory of Communication. Bell Systems Technical Journal, Vol. 27, pp. 623-656.

[Shannon, 1951]: Shannon, C.E. (1951). Prediction and Entropy of Printed English. Bell Systems Technical Journal, Vol. 30, pp. 50-64.

[Shannon et al., 1949]: Shannon, C.E. and Weaver, W. (1949). The Mathematical Theory of Communication. Univ. of Illinois Press.

[Sharif et al., 1991]: Sharif Heger, A. and Koen, B.V. (1991). KNOWBOT: an adaptive data base interface. Nuclear Science and Engineering. Vol. 107, No. 2, pp. 142-157.

[Shaw, 1993]: Shaw, J. (1993). IntelligenceWare Inc. Neural / Query Search Software. AI Expert. Vol. 8, No. 2. pp. 50.

[Shinghal et al., 1979a]: Shinghal, R., Toussaint, G.T. (1979). A Bottom-Up and Top-Down Approach to Using Context in text Recognition. International Journal of Man-Machine Studies, 1979, pp. 201-212.

[Shinghal et al., 1979b]: Shinghal, R, Toussaint, G.T. (1979). Experiments in Text Recognition with the modified Viterbi Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 184-193.

[Siedlecki et al., 1988]: Siedlecki, W., Siedlecki, K., and Sklansky, J. (1988). An Overview of Mapping Techniques for Exploratory Pattern Analysis. Pattern Recognition, Vol. 21, No. 5, pp. 411-429.

[Simpson, 1993]: Simpson, D. (1993). How Document Imaging Saves paper, Time and Money. Digital News & Review. Vol. 10, No. 11. pp. 67-71.

[Small et al., 1974]: Small, H. and Griffith, B.C. (1974). The Structure of Scientific Literature I: Identifying and Graphing Specialities. Social Studies, Vol. 4, pp. 17-40.

[Smolensky, 1987]: Smolensky, P. (1987). Connectionist AI, Symbolic AI, and the Brain. Artificial Intelligence Review, Vol. 1, no. 2, pp. 95-109.

[Sparck Jones, 1991]: Sparck-Jones, K. (1991). Journal of the American Society for Information Science. Vol. 42, No. 8. pp. 558-565.

[Sparck Jones, 1971]: Sparck Jones, K. (1971). Automatic Keyword Classification for Information Retrieval, Butterworths.

[Srihari, 1985]: Srihari, S.N. (1985). Computer Text Recognition and Error Correction: a Tutorial. IEEE Computer Society.

[Srihari et al., 1983]: Srihari, S.N. and Hull, J.J. (1983). Integrating Diverse Knowledge Sources in Text Recognition. ACM Transactions on Office Information Systems, Jan. 1983, pp. 68-87, pp. 216-235.

[Stafylopatis et al., 1992]: Stafylopatis, A. and Likas, A. (1992). Pictorial Information Retrieval Using the Random Neural Network. IEEE Transactions on Software Engineering. Vol. 18, No. 7. July 1992. pp. 590-601.

[Starks et al., 1991]: Starks, S.A. and Elizandro, D.W. (1991). Neural nets and adaptive processing. CoED. Vol. 1, No. 3, pp. 50-52.

[Stoddard, 1992]: Stoddard, B.C. (1992). Feds Scan a Prosperous Future for Electronic Imaging. Government Computer News. Vol. 11, No. 19. September 14, 1992.

[Stolcke, 1990]: Stolcke, A. (1990). Learning Feature-Based Semantics with Simple Recurrent Networks. Technical Report TR-90-015, april 1990, ICSI Berkeley, CA.

[Sun et al., 1991]: Sun, W., Liu, L.M., Zhang, W. and Comfort, J.C. (1991). Intelligent OCR Processing. Journal of the American Society for Information Science. Vol 43, No. 6, pp. 422-431.

[Sunthankar, 1992]: .Sunthankar, S. (1992). A neural network approach to text processing. Proceedings of the SPIE - The International Society for Optical Engineering. Vol. 1661, pp. 281-289. Machine Vision Applications in Character Recognition and Industrial Inspection. 10-12 Feb. 1992. San Jose, CA.

[Sutton et al., 1981]: Sutton, R.S. and Barto, A.G. (1981). Towards a Modern Theory of Adaptive Networks: Expectation and Prediction. Psychological Review, Vol. 88, pp. 135-170.

[Taghva et al., 1993]: Taghva, K., Borsack, J., Condit, A., and Erva, S. (1993). The Effects of Noisy Data on Text Retrieval. Journal of the American Society for Information Science. Vol. 45, No. 1. pp. 50-58.

[Teufel et al., 1988]: Teufel, Bernd and Stephanie Schmidt, 1988, "Full Text Retrieval Based on Syntactic Similarities", Information Systems, Vol.13(1), p.65-70.

[Touretzky, 1986]: Touretzky, D.S. (1986). BolzCONS. Proceedings of the 8th Annual Conference of the Cognitive Science Society, pp. 522-530.

[Touretzky, 1987]: Touretzky, D.S. (1987). Representing Conceptual Structures in a Neural Network. Proceedings of the IEEE International Neural Network Conference, June 21-24, San Diego, Vol. 2, pp. 279-286.

[Touretzky et al., 1988]: Touretzky, D.S. and Hinton, G.E. (1988). A Distributed Connectionist Production System. Cognitive Science, Vol. 12, Number 3, September, pp. 423-466.

[Van Opdorp et al., 1991]: Van Opdorp, G.J. (1991). Networks at Work: A Connectionist Approach to Non-Deductive Legal Reasoning. Proceedingsof the International Conference on Artificial Intelligence in Law.

[Voorhees, 1985]: Voorhees, E.M. (1985). The Cluster Hypothesis Revisited. Proceedings of the 8th Annual International ACM-SIGIR Conference on Research & Development in Information Retrieval, June '85, pp. 188-196.

[Wagner et al., 1974]: Wagner, R.A. and Fischer, M.J. (1974). The String-to-String Correction Problem. Journal of the ACM, Vol. 21, No. 1, pp. 168-173.

[Wall Street, 1991]: The First International Conference on Artificial Intelligence on Wall Street (Cat. No.91TH0399-6). 9-11 Oct. 1991. New York, NY.

[Waltz et al., 1984]: Waltz, D.L. and Pollack, J.B. (1984). Parallel Interpretation of Natural Language. Proceedings of the International Conference on Fifth Generation Computer Systems 1984, ICOT.

[Waltz et al., 1985]: Waltz, D.L. and Pollack, J.B. (1985). Massively Parallel Parsing. Cognitive Science, Vol. 9, Number 1, pp. 51-74.

[Wei et al., 1991]: Wei Li, Lee, B., Krausz, F., and Sahin, K. (1991). Text classification by a neural network. Proceedings of the 1991 Summer Computer Simulation Conference. Twenty-Third Annual Summer Computer Simulation Conference. pp. 313-318. 22-24 July 1991 Baltimore, MD.

[Weijters, 1990]: Weijters, A.J.M.M. (1990). NetSpraak: A Grapheme-to-Phoneme Conversion Network for Dutch. Proceedings of the IEEE Symposium on Neural Networks, June 21, Delft, Netherlands, pp. 59-68.

[Werbos, 1974]: Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Doctoral Dissertation Applied Mathematics, Harvard University.

[Wermter, 1991]: Wermter, S. (1991). Learning to Classify Neural Language Titles in a Recurrent Connectionist Model. In: Artificial Neural Networks (T. Kohonen, K. Makisara, O. Simula and J. Kangas, Eds.), Vol. 2, pp. 1715-1718. Elsevier Science Publishers.

[Wettler et al., 1989]: Wettler, M., Rapp, R. (1989). A Connectionist System to Simulate Lexical Decisions in Information Retrieval. In: Connectionism in Perspective (R. Pfeiffer et al., Eds.), pp. 463-469. North-Holland.

[Wettler et al., 1990]: Wettler, M., Rapp, R. (1990). Parallel Associative Processes in Information Retrieval. In: Parallel Processing in Neural Systems and Computers (R. Eckmiller, G. Hartmann and G. Hauske, Eds), pp. 509-512. Elsevier Science Publisher

[Wettler et al., 1990]: Wettler, M. and Rapp, R. (1990). Parallel associative processes in information retrieval. Parallel Processing in Neural Systems and Computers. pp. 509-512. 19-21 March 1990. Dusseldorf, West Germany.

[Widrow et al., 1994]: Widrow, B., Rumelhart, D.E., and Lehr, M.A. (1994). Neural Networks: Applications in Industry, Business and Science. Vol. 37, No. 3, pp. 93-105.

[Widrow et al., 1960]: Widrow, B. and Hoff, M.E. (1960). Adaptive Switching Circuits. 1960 WESCON Convention, Record Part IV, pp. 96-104.

[Wilbert, 1991]: Wilbert, R. (1991). Associative representation of concepts in neuronal networks. The problem of an associative access to information retrieval systems. Nachrichten fur Dokumentation. Vol. 42, No. 3, pp. 205-211.

[Wilkinson et al., 1992]: Wilkinson, R., Hingston, P. and Osborn, T. (1992). Incorporating the Vector Space Model in a Neural Network Used for Document Retrieval. Library Hi-Tech, Vol. 10, No. 1-2, pp. 69-75.

[Willett, 1984]: Willett, P. (1984). A Note on the Use of Nearest Neighbors for Implementing Single Linkage Document Classification. Journal of the ASIS, Vol. 35, No. 3, pp. 149-152.

[Willett, 1988]: Willett, P. (1988). Recent Trends in Hierarchical Document Clustering: A Critical Review. Information Processing and Management, Vol. 24, No. 5, pp. 577-597.

[Williams et al., 1989a]: Williams, R.J. and Zipser, D. (1989). Experimental Analysis of the Real-Time Recurrent Learning Algorithm. Connection Science, Vol. 1, No. 1, pp. 87-111.

[Williams et al., 1989b]: Williams, R.J. and Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation, Vol. 1, pp. 270-280.

[Winston, 1975]: Winston, P.H. (1975). Learning Structural Descriptions from Examples. In: The Psychology of Computer Vision, (P.H. Winston, Editor), McGraw-Hill Book Company.

[Wong et al., 1991]: Wong, S.K.M., Yao, Y.Y., Salton, G., and Buckley, C. (1991). Journal of the American Society for Information Science. Vol. 42, No. 10, pp. 723-730.

[Wong et al., 1993]: Wong, S.K.M., Cai, Y.J., and Yao, Y.Y. (1993). Computation of term associations by a neural network. SIGIR Forum spec. issue. pp. 107-115. Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference 27 June-1 July 1993. Pittsburgh, PA.

[Woods, 1970]: Woods, W.A. (1970). Transition Network Grammars for Natural Language Analysis. Communications of the ACM, Vol. 3 nr. 10, pp. 591-606.

[Wu et al., 1991]: Wu, S. and Manber, U. (1991). Fast Text Searching with Errors. Technical Report 91-11. Department of Computer Science, University of Arizona, Tucson, AZ.

[Zavrel, 1995]: Zavrel, J. (1995). Neural Information Retrieval: An Experimental Study of Clustering and Browsing of Document Collections with Neural Networks, Masters Thesis, University of Amsterdam, Department of Computational Linguistics, Faculty of Arts, Amsterdam, The Netherlands.

Annex: Product and Project References

Associative Dialogue System (ASDIS)

Kratzer Automatisierung GmbH
H. Geiger, R. Deffner & T. Waschulzik
Maxfeldhof 5-6
D-8044 Unterschleissheim
GERMANY
Tel: +49.89.310.2037

Bibliothekscentrum

Bibliothekscentrum
D. Quast
Vaxjo
P.O. Box 113
S-351 04 Vaxjo
SWEDEN
Tel: 0470-40045
Fax: 0470-45078

CONET-IR

University of Georgia
Department of Computer Science
G.V. Meghabghob & D.M. Bilal
Valdorta, Georgia 31698
USA
Fax: +1.921.333.7408

ELVIRA

May 3-5 1994.
De Montfort University Milton Keynes, UK.
The Gateway, Leicester
Tel: +44.533.577355
Fax: +44.533.577533

Excalibur Systems, Inc.

Excalibur Systems, Inc.
27281 Las Ramblas, Ste. 155
Mission Viejo, CA 92691
Tel: +1.800.932.9320

Global Information Management

GIM International Relations
Dan Schreirer
65 Rue de Lausanne
1202 Switserland
Tel: +41.22.738.1188
Fax: +41.22.738.1190

Image Transformation & Retrieval

Brunel University
Department of Electrical & Electronic Engineering
R. Rickman & J. Stonham
Uxbridge
Middlesex, UK

Info Select

Micro Logic Corp.
89 Leuning Street
PO Box 70
South Hackensack, NJ 07602
USA
Tel: +1.800.342.5930 or +1.201.342.6518
Fax: +1.201.342.0370

Magic Solutions, Inc.

180 Franklin Tpk.
Mahwah, NJ 07430
USA
Tel: +1.201. 529.5533

Mailfiler

Digital Apeldoorn..
Pim van der Eijk.
Ratelaar 38,
3434 EW Nieuwegein.
Tel: +31.3402-89307.
E-mail: eijk@cec.uto.dec.com.

Nestor

Nestor, Inc.
One Richmond Square
Providence, RI 02906 USA
tel: +1.401.331.9640
fax: +1.401.331.7319

Top of Mind help Desk for Windows

The Molloy Group, Inc.
90 E. Halsey Rd. Ste. 115
Parsippany, NJ 07054
tel: +1.201.884.2040
fax: +1.201.877.9177

University of Central Queensland

University of Central Queeensland
Garry Hall
Rockhampton
Queensland
Australia
Tel: 079-309.345
Fax: 079-361.361
E-mail: G.Hall@ucq.edu.all

ZyFILTER

ZyLAB Europe BV
Hoogoorddreef 9
1101 BA AMSTERDAM
The Netherlands
Tel: +31 20 691 9550
Fax: +31 20 696 5175

ZyIMAGE

ZyLAB Europe BV
Hoogoorddreef 9
1101 BA AMSTERDAM
The Netherlands
Tel: +31 20 691 9550
Fax: +31 20 696 5175

The Communities research and development information service CORDIS

A vital part of your programme's dissemination strategy

CORDIS is the information service set up under the VALUE programme to give quick and easy access to information on European Community research programmes. It is available free-of-charge online via the European Commission host organization (ECHO), and now also on a newly released CD-ROM.

CORDIS offers the European R&D community:

— a comprehensive up-to-date view of EC R&TD activities, through a set of databases and related services,

— quick and easy access to information on EC research programmes and results,

— a continuously evolving Commission service tailored to the needs of the research community and industry,

— full user support, including documentation, training and the CORDIS help desk.

The CORDIS Databases are:

R&TD-programmes – R&TD-projects – R&TD-partners – R&TD-results
R&TD-publications – R&TD-comdocuments – R&TD-acronyms – R&TD-news

Make sure your programme gains the maximum benefit from CORDIS

— Inform the CORDIS unit of your programme initiatives,

— contribute information regularly to CORDIS databases such as R&TD-news, R&TD-publications and R&TD-programmes,

— use CORDIS databases, such as R&TD-partners, in the implementation of your programme,

— consult CORDIS for up-to-date information on other programmes relevant to your activities,

— inform your programme participants about CORDIS and the importance of their contribution to the service as well as the benefits which they will derive from it,

— contribute to the evolution of CORDIS by sending your comments on the service to the CORDIS Unit.

For more information about contributing to CORDIS,
contact the DG XIII CORDIS Unit

Brussels
Ms I. Vounakis
Tel. +(32) 2 299 0464
Fax +(32) 2 299 0467

Luxembourg
M. B. Niessen
Tel. +(352) 4301 33638
Fax +(352) 4301 34989

To register for online access to CORDIS, contact:

ECHO Customer Service
BP 2373
L-1023 Luxembourg
Tel. +(352) 3498 1240
Fax +(352) 3498 1248

If you are already an ECHO user, please mention your customer number.

European Commission

EUR 16264 — Artificial neural networks for information retrieval in a libraries context

By Dr ir Johannes C. Scholtes

Luxembourg: Office for Official Publications of the European Communities

1995 — XIV, 303 pp., num. tab., fig. — 21.0 × 29.7 cm

Libraries in the information society series

ISBN 92-827-4690-9

Price (excluding VAT) in Luxembourg: ECU 33

Despite the theoretical and practical evidence that ANNs are good tools for pattern recognition tasks, it was still an open question whether they were appropriate tools within the specific domain of bibliographic information retrieval. Apart from some minor studies, no real attempt has been made to integrate an ANN as a main component of bibliographical information retrieval systems on an online public access catalogue (OPAC).

This study provides a state of the art of the application of ANN technology to information retrieval (IR) with particular emphasis on bibliographic information in a libraries context and it assesses the quality of ANN-based approaches to IR.

Venta • Salg • Verkauf • Πωλήσεις • Sales • Vente • Vendita • Verkoop • Venda • Myynti • Försäljning

BELGIQUE / BELGIË

Moniteur belge/
Belgisch Staatsblad

Rue de Louvain 42/Leuvenseweg 42
B-1000 Bruxelles/B-1000 Brussel
Tél. (02) 512 00 26
Fax (02) 511 01 84

Jean De Lannoy

Avenue du Roi 202/Koningslaan 202
B-1060 Bruxelles/B-1060 Brussel
Tél. (02) 538 51 69
Fax (02) 538 08 41

Autres distributeurs/
Overige verkooppunten:

Librairie européenne/
Europese boekhandel

Rue de la Loi 244/Wetstraat 244
B-1040 Bruxelles/B-1040 Brussel
Tél. (02) 231 04 35
Fax (02) 735 08 60

Document delivery:

Credoc

Rue de la Montagne 34/Bergstraat 34
Boîte 11/Bus 11
B-1000 Bruxelles/B-1000 Brussel
Tél. (02) 511 69 41
Fax (02) 513 31 95

DANMARK

J. H. Schultz Information A/S

Herstedvang 10-12
DK-2620 Albertslund
Tlf. 43 63 23 00
Fax (Sales) 43 63 19 69
Fax (Management) 43 63 19 49

DEUTSCHLAND

Bundesanzeiger Verlag

Postfach 10 05 34
D-50445 Köln
Tel. (02 21) 20 29-0
Fax (02 21) 2 02 92 78

GREECE/ΕΛΛΑΔΑ

G.C. Eleftheroudakis SA

International Bookstore
Nikis Street 4
GR-10563 Athens
Tel. (01) 322 63 23
Fax 323 98 21

ESPAÑA

Mundi-Prensa Libros, SA

Castelló, 37
E-28001 Madrid
Tel. (91) 431 33 99 (Libros)
431 32 22 (Suscripciones)
435 36 37 (Dirección)
Fax (91) 575 39 98

Boletín Oficial del Estado

Trafalgar, 27-29
E-28071 Madrid
Tel. (91) 538 22 95
Fax (91) 538 23 49

Sucursal:

Librería Internacional AEDOS

Consejo de Ciento, 391
E-08009 Barcelona
Tel. (93) 488 34 92
Fax (93) 487 76 59

Librería de la Generalitat
de Catalunya

Rambla dels Estudis, 118 (Palau Moja)
E-08002 Barcelona
Tel. (93) 302 68 35
Tel. (93) 302 64 62
Fax (93) 302 12 99

FRANCE

Journal officiel
Service des publications
des Communautés européennes

26, rue Desaix
F-75727 Paris Cedex 15
Tél. (1) 40 58 77 01/31
Fax (1) 40 58 77 00

IRELAND

Government Supplies Agency

4-5 Harcourt Road
Dublin 2
Tel. (1) 66 13 111
Fax (1) 47 52 760

ITALIA

Licosa SpA

Via Duca di Calabria 1/1
Casella postale 552
I-50125 Firenze
Tel. (055) 64 54 15
Fax 64 12 57

GRAND-DUCHÉ DE LUXEMBOURG

Messageries du livre

5, rue Raiffeisen
L-2411 Luxembourg
Tél. 40 10 20
Fax 49 06 61

NEDERLAND

SDU Servicecentrum Uitgeverijen

Postbus 20014
2500 EA 's-Gravenhage
Tel. (070) 37 89 880
Fax (070) 37 89 783

ÖSTERREICH

Manz'sche Verlags-
und Universitätsbuchhandlung

Kohlmarkt 16
A-1014 Wien
Tel. (1) 531 610
Fax (1) 531 61-181

Document delivery:

Wirtschaftskammer

Wiedner Hauptstraße
A-1045 Wien
Tel. (0222) 50105-4356
Fax (0222) 50206-297

PORTUGAL

Imprensa Nacional — Casa da Moeda, EP

Rua Marquês Sá da Bandeira, 16-A
P-1099 Lisboa Codex
Tel. (01) 353 03 99
Fax (01) 353 02 94/384 01 32

Distribuidora de Livros
Bertrand, Ld.ª

Grupo Bertrand, SA

Rua das Terras dos Vales, 4-A
Apartado 37
P-2700 Amadora Codex
Tel. (01) 49 59 050
Fax 49 60 255

SUOMI/FINLAND

Akateeminen Kirjakauppa

Akademiska Bokhandeln
Pohjoisesplanadi 39 / Norra esplanaden 39
PL / PB 128
FIN-00101 Helsinki / Helsingfors
Tel. (90) 121 4322
Fax (90) 121 44 35

SVERIGE

BTJ AB

Traktorvägen 11
Box 200
S-221 00 Lund
Tel. (046) 18 00 00
Fax (046) 18 01 25

UNITED KINGDOM

HMSO Books (Agency section)

HMSO Publications Centre
51 Nine Elms Lane
London SW8 5DR
Tel. (0171) 873 9090
Fax (0171) 873 8463

ICELAND

BOKABUD
LARUSAR BLÖNDAL

Skólavördustíg, 2
IS-101 Reykjavik
Tel. 551 56 50
Fax 552 55 60

NORGE

NIC Info a/s

Boks 6512 Etterstad
0606 Oslo
Tel. (22) 57 33 34
Fax (22) 68 19 01

SCHWEIZ/SUISSE/SVIZZERA

OSEC

Stampfenbachstraße 85
CH-8035 Zürich
Tel. (01) 365 54 49
Fax (01) 365 54 11

BĂLGARIJA

Europress Klassica BK Ltd

66, bd Vitosha
BG-1463 Sofia
Tel./Fax (2) 52 74 75

ČESKÁ REPUBLIKA

NIS ČR

Havelkova 22
CZ-130 00 Praha 3
Tel./Fax (2) 24 22 94 33

HRVATSKA

Mediatrade

P. Hatza 1
HR-4100 Zagreb
Tel./Fax (041) 43 03 92

MAGYARORSZÁG

Euro-Info-Service

Europá Ház
Margitsziget
H-1138 Budapest
Tel./Fax (1) 111 60 61, (1) 111 62 16

POLSKA

Business Foundation

ul. Krucza 38/42
PL-00-512 Warszawa
Tel. (2) 621 99 93, 628 28 82
International Fax&Phone (0-39) 12 00 77

ROMÂNIA

Euromedia

65, Strada Dionisie Lupu
RO-70184 Bucuresti
Tel./Fax 1-31 29 646

RUSSIA

CCEC

9,60-letiya Oktyabrya Avenue
117312 Moscow
Tel./Fax (095) 135 52 27

SLOVAKIA

Slovak Technical
Library

Nàm. slobody 19
SLO-812 23 Bratislava 1
Tel. (7) 52 204 52
Fax (7) 52 957 85

CYPRUS

Cyprus Chamber of Commerce
and Industry

Chamber Building
38 Grivas Dhigenis Ave
3 Deligiorgis Street
PO Box 1455
Nicosia
Tel. (2) 44 95 00, 46 23 12
Fax (2) 36 10 44

MALTA

Miller Distributors Ltd

PO Box 25
Malta International Airport LQA 05 Malta
Tel. 66 44 88
Fax 67 67 99

TÜRKIYE

Pres AS

Dünya Infotel
TR-80050 Tünel-Istanbul
Tel. (1) 251 91 90/251 96 96
Fax (1) 251 91 97

ISRAEL

Roy International

17, Shimon Hatarssi Street
P.O.B. 13056
61130 Tel Aviv
Tel. (3) 546 14 23
Fax (3) 546 14 42

Sub-agent for the Palestinian Authority:

INDEX Information Services

PO Box 19502
Jerusalem
Tel. (2) 27 16 34
Fax (2) 27 12 19

EGYPT/
MIDDLE EAST

Middle East Observer

41 Sherif St.
Cairo
Tel/Fax (2) 393 97 32

UNITED STATES OF AMERICA/
CANADA

UNIPUB

4611-F Assembly Drive
Lanham, MD 20706-4391
Tel. Toll Free (800) 274 48 88
Fax (301) 459 00 56

CANADA

Subscriptions only
Uniquement abonnements

Renouf Publishing Co. Ltd

1294 Algoma Road
Ottawa, Ontario K1B 3W8
Tel. (613) 741 43 33
Fax (613) 741 54 39

AUSTRALIA

Hunter Publications

58A Gipps Street
Collingwood
Victoria 3066
Tel. (3) 9417 53 61
Fax (3) 9419 71 54

JAPAN

Procurement Services Int. (PSI-Japan)

Kyoku Dome Postal Code 102
Tokyo Kojimachi Post Office
Tel. (03) 32 34 69 21
Fax (03) 32 34 69 15

Sub-agent:

Kinokuniya Company Ltd
Journal Department

PO Box 55 Chitose
Tokyo 156
Tel. (03) 34 39-0124

SOUTH and EAST ASIA

Legal Library Services Ltd

Orchard
PO Box 0523
Singapore 9123
Tel. 243 24 98
Fax 243 24 79

SOUTH AFRICA

Safto

5th Floor, Export House
Cnr Maude & West Streets
Sandton 2146
Tel. (011) 883-3737
Fax (011) 883-6569

ANDERE LÄNDER
OTHER COUNTRIES
AUTRES PAYS

Office des publications officielles
des Communautés européennes

2, rue Mercier
L-2985 Luxembourg
Tél. 29 29-1
Télex PUBOF LU 1324 b
Fax 48 85 73, 48 68 17

15

CD-NA-16264-EN-C

Results of a study done
by MSC Information Retrieval Technologies BV,
the Netherlands, for the European Commission,
Telematics for libraries

NOTICE TO THE READER

All scientific and technical reports published by the European Commission are announced in the monthly periodical **'euro abstracts'**.
For subscription (1 year: ECU 63)
please write to the address below.

Price (including VAT) in Luxembourg: ECU 33

ISBN 92-827-4690-9

OFFICE FOR OFFICIAL PUBLICATIONS
OF THE EUROPEAN COMMUNITIES

L-2985 Luxembourg

9 789282 746905