Detecting Anomalous Events over Time Using RDF Triple Extraction and a Dynamic Implementation of OddBall

Benedikt Heinrichs and Jan C. Scholtes
Department of Data Science and Knowledge Engineering, Maastricht University

***Abstract*— This paper shows a new approach for anomaly detection by combining the extraction of so-called triples consisting of a subject, predicate, and object using dynamic anomaly-detection. First, the methods used to extract triples and general principles of anomaly detection and event detection are discussed. Next, a novel approach is presented where extracted triples are converted into time-lapsed networks of triples on which anomaly and event detection methods from social network analysis are applied. Subsequently, the results of the experiments are presented together with the evaluation method used. Considering the results of the tested methods, the dynamic variation of the OddBall algorithm, which measures network changes over time, displays the connection between the predictions of our model and real-life events accurately.**

I. Background

Identifying new and important events from textual information, is an easy task for humans, but a difficult one for computer programs. Humans will immediately recognize new and changing events as anomalous. For an algorithm, it is difficult to distinguish between normal information and anomalous information, especially since anomalous information exists in many forms, shapes and is highly-context sensitive.

This study aimed to create an algorithm that has the ability to detect anomalous events without additional domain-specific or other forms of background knowledge. The method used in this paper, extracts subjects, predicates and objects from text. These information holders are used as so-called triples that can be converted into dynamic network structures from which anomalies can be detected. Due to the lack of annotated corpora, the methods have not been exhaustively tested yet. However, initial experiments on real-world data, show a promising ability to detect anomalous events.

II. Introduction

In this section, an overview is given of the text corpus, the triple-extraction method and the anomaly-detection methods used in this study.

A. Reuters text corpus

The text corpus used for this paper is the Reuters RCV1 described by [1], which contains over 800,000 news wire stories from August 20, 1996 to August 19, 1997. This corpus is a well-known and often used data set for text-mining research. In addition, the news content was considered to fit to our research, because the content of news is changing day by day, and important new events can be seen as anomalous. We used information from Wikipedia to identify a selection of important events and set up experiments to determine whether our detection methods could identify these events.

B. Resource Description Framework

The Resource Description Framework (RDF), as defined by [2], and is a model that describes data by triples including a subject, predicate, and object. An example of such triples can be seen in figure 1 where the linking from subject to predicate and predicate to object is shown. The example describes that George Washington was a president.

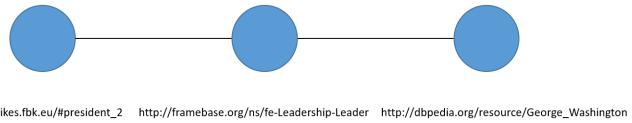

Fig. 1. An example of Resource Description Framework triples displayed as a graph

These subjects, predicates, and objects are denoted by a Uniform Resource Identifier (URI), which links to the definition of the term, for example, stating that an object or subject is a person. For this reason, RDFs are useful to represent semantic relations in textual information. The structure of a RDF model allows sets of RDFs to be converted into a graph, where the object and subject are represented as nodes and the predicates as vertices, resulting in a network of semantic relations for a textual document.

C. Extracting triples from text

Triples including a subject, predicate, and object can be extracted from text by understanding the grammar and dependencies of the given text. There are several existing methods to handle triple extraction.

1) Stanford CoreNLP: [3] describes Stanford CoreNLP, a pipeline for applying text-analysis tools to plain text. The toolkit features popular methods like part-of-speech (POS) to assign syntactic roles, such as noun or verb. The toolkit is also capable of named- entity recognition to label tokens with their semantic role, such as names of persons and locations. The Stanford Open Information Extraction, based on the above core functionality, extracts relation tuples, typically binary relations, from plain text. These relations do not need to be specified in advance. It implements the system proposed by [4] and converts, for example, the text “Barack Obama was born in Hawaii” to the subject “Barack Obama”, the predicate “was born in”, and the object “Hawaii”. These are

similar to our concept of triples, although the extracted tuples only provide a superficial description.

2) Extraction of RDF triples: In section II-B on the previous page the RDF format is defined. Given a certain text, content and meaning can be represented in RDF format. There exist various tools to convert simple text into the RDF format. The Pikes library [5], combines several text-mining methods to create semantically-rich descriptions of textual documents. With these it creates knowledge directly from text. This property was deemed important in this project; therefore, extraction by Pikes was chosen over extraction by Stanford CoreNLP.

D. Anomaly detection

All data sets can be grouped into subsets containing similar objects, and these subsets can then be labeled. In large, natural data sets, there are always objects that cannot be assigned to any of the existing subgroups because they are too different, based on a predefined similarity measure; these objects are called anomalies. For example, a large number of news articles reference that Barack Obama was born in Hawaii. An individual news article stating that Barack Obama was born in Kenya would be considered anomalous, since it does not conform to previously established and frequently mentioned facts. Chandola et al. [6], provide a good overview of several anomaly detection methods for different data types and domains. The most interesting ones for the textual domain are as follows.

1) OddBall: In [7], the researchers proposed a new algorithm to detect anomalies, called OddBall, which uses a number of power laws. It is based on the intuition that there are a number of natural distributions that follow such power laws. An example is the usage of words in natural language, where very few words are used highly frequently, and many words are used rarely. The OddBall algorithm presumes a similar power-law distribution between the nodes and vertices of social networks. As the Object-Predicate-Subject networks that we construct in this study originate from natural language, we presume that our networks could also be subject to such power laws, in which case it makes sense to use an algorithm such as OddBall anomaly detection.
The OddBall algorithm focuses on the immediate neighborhood of each node. The subgraph constructed from these neighboring nodes is called an *egonet*. It is proposed that, if the neighborhood of a node differs from the others, the node is anomalous. The researchers also proposed the following formula for scoring anomalies:

$$out\text{-}line(i) = \frac{max(y_i, Cx_i^\theta)}{min(y_i, Cx_i^\theta)} * log(|y_i - Cx_i^\theta| + 1).$$

The formula basically calculates the distance of a node i to a fitting line. The values can be understood as the distance of the points x_i and y_i for a node i to the expected value given by the power law equation $y = Cx^\theta$. Therefore the distance of y_i to Cx_i^θ defines the outlier score. Based on these power laws and the scoring function, anomalies can be detected with OddBall. The researchers' results show that OddBall indeed spots unusual nodes even in huge graphs.

2) NetSimile: [8] describes a method that is able to compare graphs by extracting certain features of each provided graph. Then, the features are translated into signature vectors. In the next step, these signature vectors are compared, and this comparison is used to return the similarity values between the graphs. These similarity measures are based on the so-called Canberra distance, which is a distance measure for a pair of points in a vector space. Using this technique, anomalies in graphs can be found over time when there is a unique graph for each time period. In this setup, the NetSimile algorithm can be used to identify important events over time. The researchers showed with extensive experiments, that NetSimile is superior over other graph comparison algorithms.

III. Approach

In the following, the extraction of RDF triples, the implementation of NetSimile, and the implementation of our developed method for anomaly and event detection are described.

A. Extraction of RDF triples

In section II-C.2, the tool Pikes was presented. The following will explain its usage, how text was converted into RDF format, and what novel methods were applied to filter out redundant spellings for similar natural objects to reduce the complexity of the triple graphs.

1) Implementation and usage: Using Pikes, the Reuters RCV1 corpus was converted into a large set of RDF files.

2) Extracting the most important information from RDF files: The RDF files contained a wealth of semantic information; however, not all of this information was needed to represent relations between objects and subjects. Therefore, we seleted the most relevant information. The most important relations are shown in figure 2.

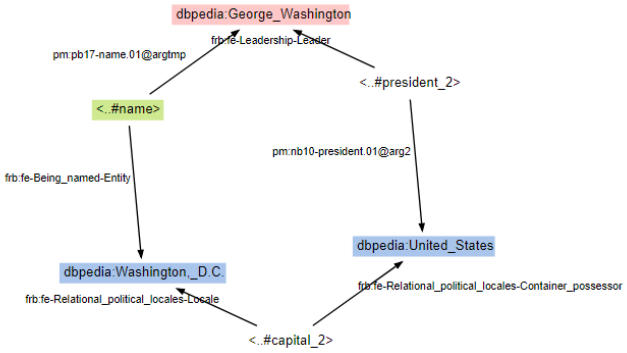

Fig. 2. Representation by Pikes of the statement "Washington, D.C. is the capital of the U.S. It was named after George Washington, the first president of the U.S."

An extraction of triples containing only the most important predicates was therefore deemed to be an important step. The results of this process will be discussed in more detail in section III-B.1 on the following page.

3) Extracting triples by date: Initial experiments on the extracted networks did not yield good results for anomaly detection. As most anomalous events occur over time, we decided to add a dynamic component and create separate networks (one per day) holding information from multiple new messages. The Reuters corpus comes with data that is linked to certain dates. This property makes it possible to create a dynamic data set where major changes in the network of relevant triples over time can be considered anomalous or important events. In order to create such a triple network, extracted triples were combined into one large RDF file by date.

4) Extracting only certain topics: While extracting every triple by date provides a wealth of information, the data sets contained too much data to identify general anomalies. This problem originated from the fact that the original large data set consisted of too many subtopics and, within each of these subtopics, there was no clear indication of what happened. We expect that the reason for this is that many different subtopics have contrasting high points; therefore, the results are not interpretative when they are mixed. This problem was tackled by focusing on certain subtopics. "Zaire" and "Congo" were used for the topic of the First Congo War and "hurricane", "blizzard" and "typhoon" were used for the topic of weather. Using these subtopics gave a clear view of the most anomalous events and allowed for an easier interpretation of detected anomalies and events.

The First Congo War went from October 24, 1996 until May 16, 1997. It started as a foreign invasion of the country Zaire led by Rwanda and ended in renaming Zaire the Democratic Republic of the Congo. For this topic, selected triples included at least one of the terms shown in table I, referencing the people and locations that are important to this topic. The weather-related terms studied are also shown in table I.

First Congo War topics	Weather topics
Congo	Weather
Zaire	Storm
White Legion	Drought
National Union for the Total Independence of Angola	Hurricane
Army for the Liberation of Rwanda	Tornado
Interahamwe	Tsunami
Rwanda	Heat wave
Uganda	Cold wave
Burundi	Cyclone
Angola	Heat burst
Mobuntu Sese Seko	Monsoon
Jonas Savimbi	Blizzard
Paul Rwarakabije	Typhoon
Robert Kajunga	Lightning
Tharcisse Renzaho	
Laurent-Desire Kabila	
Paul Kagame	
James Kabarebe	
Yoweri Museveni	
Pierre Buyoya	
Jose Eduardo dos Santos	

TABLE I

FIRST CONGO WAR AND WEATHER TOPICS

B. Representing RDF triples as a graph

RDF files can be converted into a graphical structure by considering a triple as a node pointing to another node using a predicate. By representing every triple as nodes and edges we created a large graph: every node that gets converted from the Python library RDFLib is interpreted by the URI, which defines it and different URIs; for example, "http://pikes.fbk.eu/#Zaire_3" and "http://pikes.fbk.eu/#Zaire_5" are deemed different nodes. With direct inclusion using string normalization techniques described in [9], URIs using different textual representations of the same real-world objects are normalized to one unique textual representation. Using this method, the different references to "Zaire" were normalized to one reference.

1) Extracting the most important information from RDF files: Next, the most relevant predicates were extracted from the full RDF files by using the keyword filtering. The result can be seen in figure 3 and is comparable to the original output from Pikes shown in figure 2 on the preceding page.

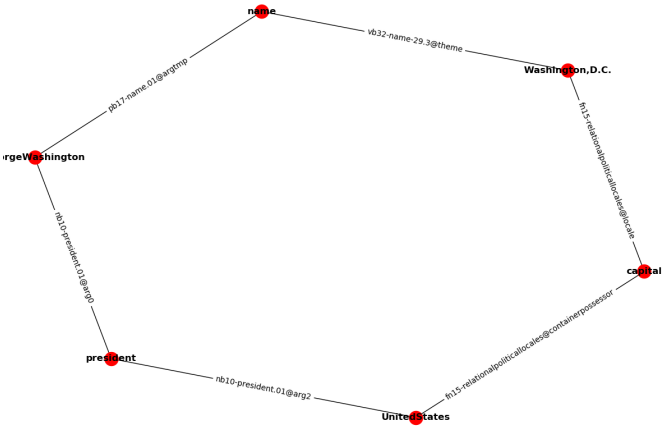

Fig. 3. Own representation of the statement "Washington, D.C. is the capital of the U.S. It was named after George Washington, the first president of the U.S."

Subsequently, these topic-focused RDF triples were used to construct a topic-focused network of Subject-Predicate-Object URIs per day.

C. Proposed anomaly detection method

In section IV-A on the next page, it will be shown that the application of the standard NetSimile algorithm did not produce the results expected on the established data set. Therefore, a novel approach for anomaly detection was developed by using the OddBall algorithm and combine it with a dynamic component to overcome the limitations of NetSimile.

1) Dynamic OddBall: Our OddBall method combines some of the previously mentioned methods. First, the RDF triples are represented as graphs. OddBall then generates the anomaly values of the most anomalous nodes in each graph. As each day is represented as an individual graph with a unique set of anomalous nodes generated by the OddBall algorithm, information on the anomalous nodes in the previous graph can be compared with one in the current graph. When nodes are anomalous in both graphs, this information is discarded; however, new anomalous nodes

in graphs from later days are kept since these might indicate a new upcoming event. The number of such new changes are then summed. The resulting value determines the dynamic anomaly score of the current graph. We decided not to normalize these values as certain anomalous events can have more impact than others, so there is no maximum for this value. The results of this approach are shown in section IV-B on the following page.

D. Evaluation methods

Evaluating the results is a difficult task due to the results not being known before retrieving them. To our knowledge, there are no annotated corpora that would allow us to conduct a full quantitative and qualitative evaluation. In order to provide some indication of quality, we decided to use a measurement for accuracy based on the correlation to anomalous and major events in real-life events. This means that anomalous points can be seen as major events or changes related to a specific topic. If these anomalous points then correlate to certain events, for example, the First Congo War, the beginning and ending of the war, we considered the result accurate.

IV. Results

In the following, the results of the algorithms are presented. Figures were created to display the results of each algorithm. For each figure, the x-axis describes the date where the detected events occurred and reaches from 0 (August 20, 1996) to 364 (August 19, 1997).

A. NetSimile

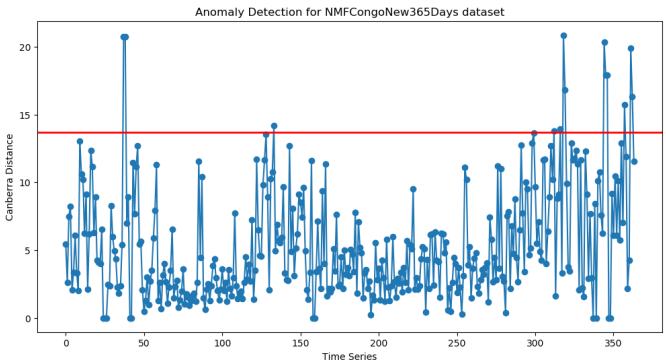

Fig. 4. NetSimile on *Zaire*

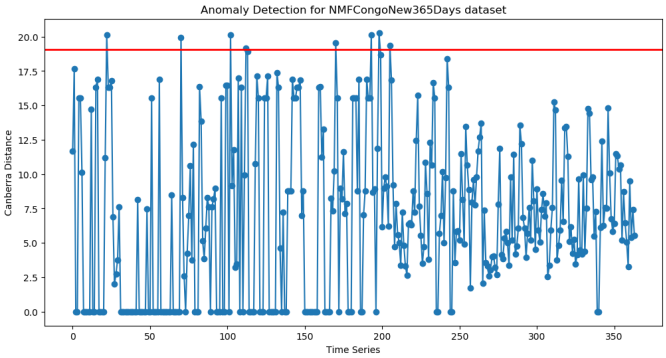

Fig. 5. NetSimile on *Congo*

1) First Congo War: The explored subtopics of the First Congo War is based on news messages for Zaire, selected by keywords as shown in figure 4 and news messages for Congo selected by keywords as shown in figure 5. For the topic of Zaire, some high and low points can be seen in the graph; however, the highest points in the graph exist at random points of no significance during the First Congo War. This is quite the opposite from the expected result. The expectation would be high points at the beginning and the end of the war. We then expected that the results from selected news messages on Congo would give better results in that the name Democratic Republic of the Congo was not used until after the First Congo War was over and we would expect a peak after the end of the war. There were no significant correlations of the high points in this graph and we did not observe the expected peaks. Furthermore, there were still many points in terms of high values before the Democratic Republic of the Congo was founded. Overall, the results for the subtopics were rather disappointing, and using the standard NetSimile algorithm did not provide a method to detect major events related to the First Congo War.

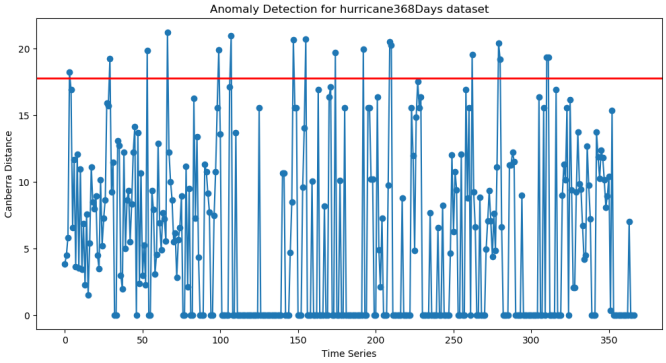

Fig. 6. NetSimile on *hurricanes*

2) Hurricanes: The figure 6 shows clusters of anomalies in three different parts. The first part covers 0-65, the second from 66-260 and the third from 320-365. During the first and third clusters, more activity can be observed in the graph than in the second cluster. When compared to real-life events and dates, these data points correlate to the Atlantic

hurricane season. The first cluster covers from end of August 1996 to November 1996, and the third from June 1997 till August 1997, both of which are hurricane seasons. Especially interesting is the fact that the representation for August 1997 does not have much activity, which also correlates to real-life events where, during August 1997, there were no hurricanes. There is still significant noise in the second cluster, which can be seen by seemingly random high points, that might originate from different entities also called “hurricane”.

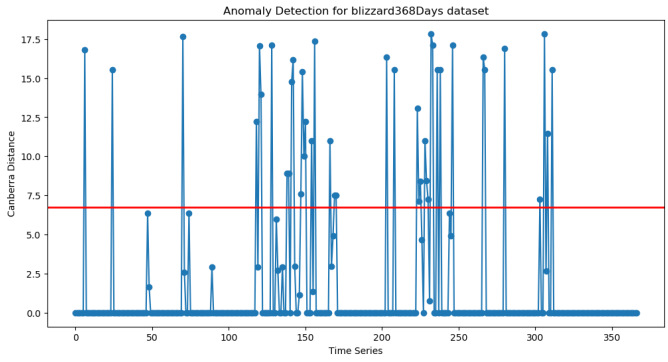

Fig. 7. NetSimile on *blizzards*

3) Blizzards: The results in figure 7 show high points in different regions. The first cluster of high points refers to December 1996-January 1997, during which blizzards occurred. The second cluster of high points refers to April 1997-May 1997, during which a blizzard occurred on April 1st. The results show suspected regions of blizzard occurrences. However, there is no clear indication as to when a blizzard happened, and it is also quite noisy. So, in this case, the standard NetSimile algorithm does not work well.

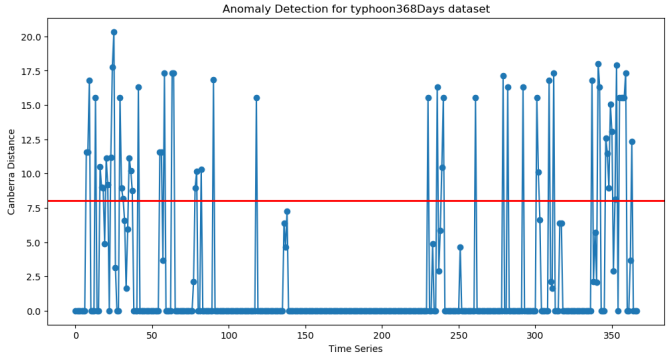

Fig. 8. NetSimile on *typhoons*

4) Typhoons: Figure 8 shows two different clusters of high points. Between these, there is a clear gap; this represents a time where no typhoon occurred, and this correlates to the typhoon seasons. The first cluster correlates to the typhoon season of 1996 and the second cluster to the one of 1997. This shows a real-life correlation in detecting typhoon seasons using the standard NetSimile algorithm.

Despite the last success, the results of applying NetSimile, show that it is not a stable and reliable method to detect anomalous events in our use case.

B. Dynamic OddBall

The results of applying dynamic OddBall to our data sets resulted in more stable and better results.

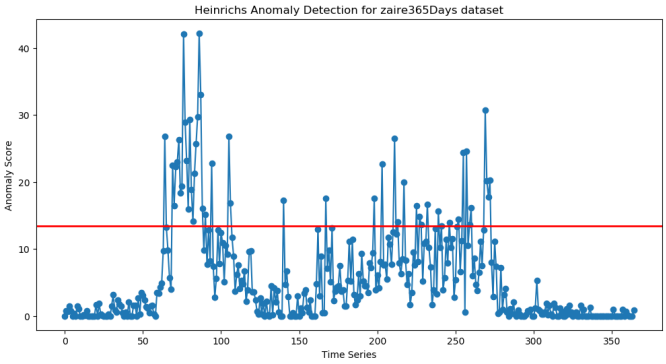

Fig. 9. Dynamic OddBall algorithm on *Zaire*

1) First Congo War: Figure 9 shows that, from 0-50 and 300-365, there are little to no spikes in the graph, which correlates to the period before the start and after the end of the First Congo war. At 65, which represents October 24, 1996, there are high spikes in the figure, which correlates to the beginning of the First Congo War. After this, it can be seen that there are regular, but smaller, anomalous events with many high spikes until around 250. From 250-280 there are spikes, and then activity fades, which correlates to the dates from April 24, 1997 to May 27, 1997 and to the end of the war, which was on May 16, 1997. As seen with this example, significant events can be detected by the dynamic OddBall anomaly detection method. Using the same methods for the First Congo War, the duration and some of the most important events of the Frst Congo War are displayed in the graph in figure 9.

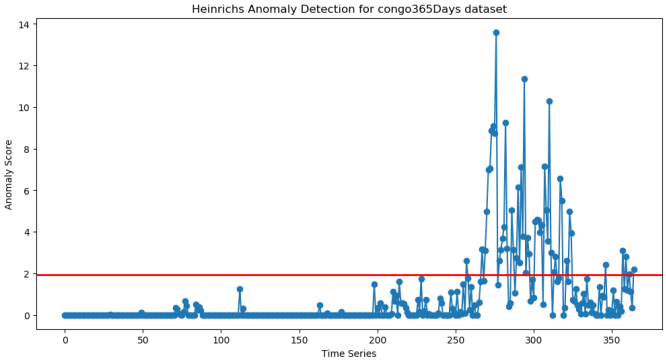

Fig. 10. Dynamic OddBall algorithm on *Congo*

Figure 10 describes Congo as a topic, which syncs up to Zaire as a topic. As can be seen by comparing figure 9 to figure 10, Congo, as a topic only became relevant after the war stopped and the Democratic Republic of the Congo was founded. From this, a correlation to real-life historic events can be seen.

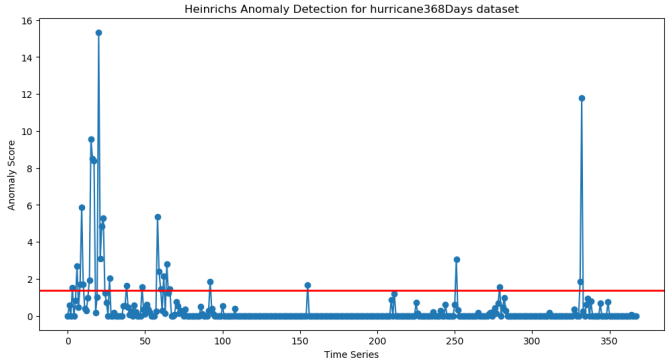

Fig. 11. Dynamic OddBall algorithm on *hurricanes*

2) Hurricanes: Figure 11 displays some high activity at the beginning of the graph and some activity during the end. The first high activity cluster, correlates directly to the hurricane season in 1996, and the last activity cluster correlates to the hurricane season in 1997. During the time series values of 150-300 there are some smaller spikes, these do not correlate to the hurricane season and might be caused by other entities called “Hurricane”.

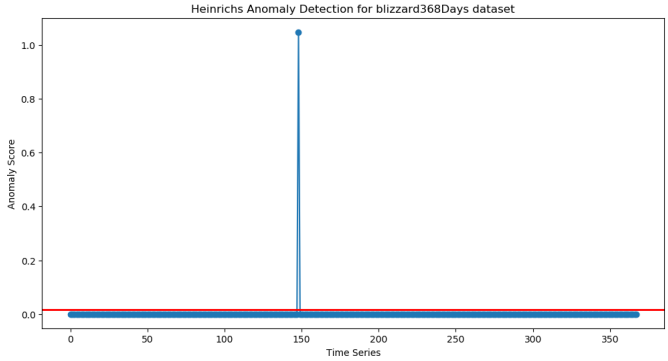

Fig. 12. Dynamic OddBall algorithm on *blizzards*

3) Blizzards: In figure 12, there is only one spike. The data set underneath, however, shows that some values in the graph are close to zero but are too small to be seen in the plot. As such they were interpreted as too insignificant to be displayed. The one larger spike correlates to real-life data: it displays a high anomaly score on January 16, 1997, when a blizzard actually occurred. Due to this property, a correlation to some real-life events can be shown. There are, however, events missing, which is considered anomalous. On the 1st of April 1997, a blizzard occurred that is not represented in the graph. In the actual values used to create the graph, there are near-zero values for April 4, 1997 and April 9, 1997 that indicate this event. However, since these values are so small, they did not appear in the news frequently and were thus not picked up by the anomaly detection method.

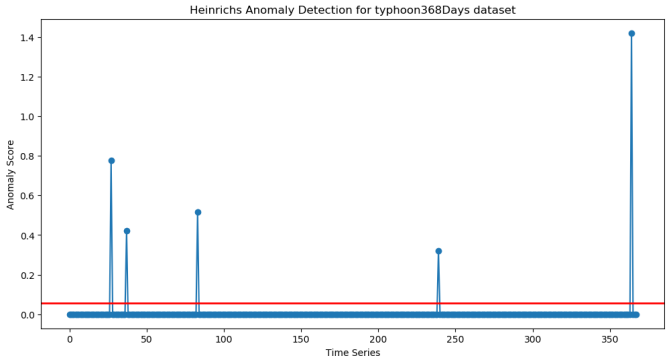

Fig. 13. Dynamic OddBall algorithm on *typhoons*

4) Typhoons: Figure 13 does not indicate the typhoon seasons. It only features five high points with more near zero-points in the data set. The high points all occured in typhoon seasons, but reviewing them did not give a clear indication as to which event they indicate. One correlation to real-life events is that no high points or near-zero points existed during non-typhoon seasons. Meaning can be only read from the two high points near the end and beginning of the typhoon seasons. The first one points to November 12, 1996, which is close to the end of the typhoon season in 1996 and the second one points to April 17, 1997, which is close to the beginning of the typhoon season in 1997.

V. Conclusion

This paper described a novel method to convert regular text into RDF files, enhancing the given text with a rich description. From these RDF files, the relevant triples were extracted by selecting predicates and extracting events, that is, the First Congo War or extreme weather. The proposed method, Dynamic OddBall, performed better than NetSimile in detecting the First Congo War when looking at the results. Instead of having random high spikes, dynamic OddBall accurately detected the start and the end of the war. When comparing both algorithms to the weather data, both perform well; however, NetSimile produced more spikes in random places, whereas dynamic OddBall displayed less spikes where they should have been present. This was especially true for the blizzard and typhoon topics, but this could be related to the amount of news coverage pertaining to extreme weather events compared to the First Congo War. It is therefore concluded that dynamic OddBall is more stable and accurate than NetSimile in this domain for the tested topics. Therefore, dynamic Oddball tentatively shows to be a promising method for anomaly and event detection.

VI. Future Work

Anomaly detection in text, especially with triples, should be developed further. Using the novel triple extraction method with a different (annotated) data set could be an interesting approach to test how well the proposed algorithm performs against other algorithms that detect changes in a network over time. Such a data set would need to include textual data organized on a time scale and intensity values

which describe how relevant a certain topic was at a certain date. If no annotated data sets become available to test our algorithms on, the generation of synthetic data sets, which is a common practice in social network research, could also provide a better indication of the stability and quality of the method proposed here.

REFERENCES

[1] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005332.1005345

[2] D. Wood, M. Lanthaler, and R. Cyganiak. (2014, Feb.) RDF 1.1 concepts and abstract syntax. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[3] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[4] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2015, pp. 344–354. [Online]. Available: http://www.aclweb.org/anthology/P15-1034

[5] F. Corcoglioniti, M. Rospocher, and A. P. Aprosio, "Frame-based ontology population with pikes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3261–3275, Dec 2016.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[7] L. Akoglu, M. McGlohon, and C. Faloutsos, "oddball: Spotting anomalies in weighted graphs," in *Advances in Knowledge Discovery and Data Mining*, M. J. Zaki, J. X. Yu, B. Ravindran, and V. Pudi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 410–421.

[8] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Netsimile: A scalable approach to size-independent network similarity," *CoRR*, vol. abs/1209.2684, 2012. [Online]. Available: http://arxiv.org/abs/1209.2684

[9] K. Borggrewe and J. Scholtes, "Domain-independent method for entity resolution by determining textual similarities with a support vector machine," *Benelux Artificial Intelligence Conference (BNAIC)*, 2016.