Towards a complex network approach to detecting events in high-volume news streams

Anonymous authors

Abstract

Detecting important events in high volume news streams is an important task for a variety of purposes. The volume and rate of online news increases the need for automated event detection methods that can operate in real time. In this paper we develop a network-based approach that makes the working assumption that important news events always involve named entities (such as persons, locations and organizations) that are linked in news articles. Our approach uses natural language processing techniques to detect these entities in a stream of news articles and then creates a time-stamped series of networks in which the detected entities are linked by co-occurrence in articles and sentences. In this prototype, weighted node degree is tracked over time and change-point detection used to locate important events. Potential events are characterized and distinguished using community detection on KeyGraphs that relate named entities and informative noun-phrases from related articles. We performed an evaluation against human annotation and against simple text-based clustering, finding that our system detects more events than simple clustering and compares well to human annotations. This methodology already produces promising results and will be extended in future to include a wider variety of complex network analysis techniques.

1. Introduction

The volume and velocity of online news has increased dramatically in recent years. For news analysts in various domains (e.g. politics, finance, open-source intelligence) this creates a need for automated methods to detect and summarize news events in real time, since doing so with human effort alone is rapidly becoming intractable. Automated methods can assist human analysts by providing alerts to emerging news events and directing the analyst towards relevant articles.

Here we present work-in-progress that aims to develop a network-based methodology for news event detection and characterization. We make the assumption that news events link multiple entities (e.g. people, organizations, places) at a particular point in time. We hypothesize that news events may therefore be effectively detected by studying the temporal
evolution of a knowledge graph that links entities based on their co-occurrence in news articles. In this paper we explore this approach by developing a prototype software pipeline that creates and analyses complex networks in which the nodes are entities (here Persons, Organizations and Locations) and the edges represent entity co-occurrence in news articles. Entities are identified using natural language processing techniques, with co-occurrence measured at both document and sentence levels, applied to a large corpus of news articles. In this version of the method, changes in node degree are used as an indicator to highlight possible news events, which are then characterized using an extended knowledge graph that incorporates noun phrases alongside entities. Future work will seek to optimize and validate this methodology. In this paper we restrict our aims to demonstrating the feasibility of the approach and providing some preliminary results that suggest it can be effective.

As a test data set we use the All_The_News data set from the Kaggle website\textsuperscript{1}. The data set contains $\sim$140,000 articles from major news sites of the United States of America such as the New York Times, Breitbart, CNN, New York Post and Reuters. We use all available articles for 2016, an eventful year which saw key political events including the Brexit referendum and the election of US President Donald Trump. We test the method by seeking to detect and identify major news events within 2016. We also perform a focused evaluation against human annotation for news events during the week of the Brexit referendum in the UK.

The paper is structured as follows: First we review related work in Section 2. In Section 3 we describe our methodology for knowledge graph creation, event detection and event characterization. In Section 4 we present some preliminary results from application of the methods to the All_The_News data set. Finally, in Section 5 we present the conclusions of our research.

2. Previous Work

The research area of topic/event detection and tracking (TDT) in news streams has a long history [Allan et al., 1998, Wayne, 1998]. TDT typically combines natural language processing and understanding (e.g. named entity recognition, part of speech tagging, entity disambiguation), information retrieval (e.g. reversed indexing of document keywords, document similarity and clustering), social network analysis (e.g. using relations between document entities to cluster documents describing the same event into network communities) and machine learning (e.g. extraction of document features to identify context and create clusters).

Event detection research in TDT has two main approaches: document pivot and feature pivot. The goal of the document pivot approach is to create document clusters that describe the same event and then extract the appropriate features from them to categorize incoming articles [Allan and Lavrenko, 1998]. The feature pivot approach focuses on detecting hidden features to cluster documents and identify news events [He et al., 2007, Weng and Lee, 2011]. TDT research can also be categorized as either new event detection (dealing with document streams processed on-line or in another live mode) and retrospective event detection (off line discovery of events in a historical corpus) [Yang et al., 1998].

\textsuperscript{1}https://www.kaggle.com/snapcrack/all-the-news
Many event detection methods are based on some form of time series analysis applied to word frequencies. Kleinberg [Kleinberg, 2002] used an infinite state automaton to model changes of word frequency in a document stream, with state transitions considered as events. Allan et al. [Allan and Lavrenko, 1998] used a modified version of TF/IDF (term frequency-inverse document frequency, a form of weighted word frequency metric) to identify events as document clusters, also weighting by the time separation between the current document and the candidate event cluster. Since future document features are not known, the online approach of this algorithm needed to estimate the IDF metric which these authors approached by using an auxiliary data set. Another approach to event detection has a focus on word correlations, which are usually measured by distributional similarity [Li and Yamanishi, 2000, Wartena and Brussee, 2008] or the number of word co-occurrences [Prabowo et al., 2008]. Fung et al. [Fung et al., 2005] worked on detecting important bursty events in text streams. Their technique detects a minimal set of features that correspond with a number of events in a given time window. The features are identified by statistically modelling the frequency of each individual word in an incoming document with a binomial distribution. Then these features are associated with events and time series analysis is used to detect significant changes (change points, peaks, troughs) which could correspond to an important event. This approach uses a large number of features and can be computationally expensive, in addition high frequency words may not be always useful for user interpretation of the content of a detected event. Another approach for event detection with Twitter data is to use recognized named entities in the text, cluster the documents (tweets) that contain each entity, then apply machine learning algorithms to decide whether the selected documents constitute an event regarding the detected entity [Popescu et al., 2011, Aiello et al., 2013]. Mele et al. [Mele and Crestani, 2017] detected events from heterogeneous news streams by combining topic modelling, named entity recognition and temporal modelling. Document clusters that describe one event are created using the most frequent named entities and event phrases of each article as features.

A few studies have incorporated network analysis into their event detection methodology. Melvin et al. [Melvin et al., 2017] introduced a new approach where they create a noun phrase network (PhraseNet) by extracting high frequency phrases from tweets and linking them by their co-occurrences. After creating the network, candidate events are identified by running a community detection algorithm using the assumption that a network community represents a potential event. To validate which candidate events are actual events their distributions over time (time series) are monitored and characteristics such as the number of peaks, intensity of the peaks and distribution variance indicate the importance of the event. The idea of utilizing a noun phrase network is also used by Sayyadi et al. [Sayyadi et al., 2009]. Their algorithm first extracts named entities and noun phrases from a news document and then creates a network of extracted terms, which they call a KeyGraph (after Ohsawa et al. [Ohsawa et al., 1998] and Mori et al. [Mori et al., 2004, 2006]). Nodes in the KeyGraph are document keywords and edges are formed when those keywords co-occur in a document. Community detection techniques are applied in order to discover events as represented by network communities.
3. Methodology

Our method assumes that the occurrence of a news event is signified by changes in prominence of three kinds of entities: Persons, Locations and Organizations. Our approach identifies such entities in a stream of news articles and then forms a sequence of networks in which the nodes are unique named entities and the edges represent entity co-occurrence within an article or sentence. News events are detected by finding peaks and change points in time series associated with individual entities; here we use weighted node degree as the key metric. The news article stream is filtered down to only retain articles referring to these “peaking” entities, based on a working assumption that these are the most news-worthy entities. An entity-phrase network is then created to help interpret the detected events, which includes noun phrases as nodes alongside the previously detected entities. Community detection is then used to find communities within the entity-phrase network, with each found community considered to represent a candidate event. Figure 1 shows the main steps of the process.

The graph-based method has two significant advantages. Firstly, the graph structure allows the importance of an entity to be measured not only using the frequency of its appearance in articles but also taking into account its relationship with other entities. Here an entity is important not only if it appears frequently but also if it co-occurs with (is linked with) other important entities. Secondly the KeyGraph (the entity co-occurrence graph extended with noun phrases) is very useful for detecting and characterising news. Here applying the Louvain algorithm [Blondel et al., 2008] divides the KeyGraph into strongly connected entity/noun-phrase communities, each corresponding to a candidate event.

![Image of the main steps of the method]

Figure 1: Diagram of the main steps of the method

3.1 Entity Detection

The goal of our approach is to detect important events, given a stream of newspaper articles or a collection of articles. For each article we apply a named entity recognition (NER) technique in order to detect three kinds of entities: Persons, Locations and Organizations.
We experimented with a number of different NER tools (specifically NLTK ne_chunk\textsuperscript{2}, Stanford NER \cite{finkel2005 dementia}, and SpaCy\textsuperscript{3}). We finally chose to work with the Stanford NER classifier because it is trained with the CoNLL\textsuperscript{4} data set which consists of Reuters newswire articles, which is ideal in our case, and because of its good performance.

After detecting named entities we run a disambiguation process. For Person entities, we replace single words (typically first or last names) with the full name of the entity to avoid spurious duplication. Each document is scanned for Person entities. When a single word Person entity is found, it is replaced by the most recent matching multi-word Person entity phrase that was found. If no match is found the single-word name is retained. An example of this is shown in Figure 2 manual checking showed that this method works well.

For Location and Organization entities, we disambiguate by expanding abbreviations and reference to manually collated dictionaries of exceptions.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Person entity name disambiguation example}
\end{figure}

\subsection*{3.2 Knowledge Graph Creation}

For creating the knowledge graph we use the detected entities as nodes and entity co-occurrence in articles and sentences to form weighted edges. In determining edge weight, we consider the importance of each entity within each article and sentence. To do this, we assign a significance value to each detected entity in a given document (which may be an article or a sentence) as follows:

\begin{equation}
S_x(v) = \frac{tf(v, x)}{\sum_{v' \in V} tf(v', x)}
\end{equation}

where \(v\) is the current entity (node), \(x\) is a piece of text (document, sentence), \(tf(v, x)\) is the term frequency (the raw count of term \(v\) in text \(x\)), and \(V\) is the set of all entities in the current document. Figure 3 shows example output from application of Equation 1.

\textsuperscript{2} http://nltk.org
\textsuperscript{3} https://spacy.io/
\textsuperscript{4} http://www.conll.org/
The contribution from an article $d$ to the edge weight joining two entity nodes $i, j$ is then given by:

$$w_d(i, j) = \begin{cases} S_d(i) + S_d(j) + \sum_{r \in d}(S_r(i) + S_r(j)) & \text{if } i, j \in V \\ 0 & \text{otherwise} \end{cases}$$

(2)

where $V$ is the set of entities in the current document and $r$ is a unique sentence in that document.

Thus by Equation 2 we take into account two kinds of co-occurrence, when two entities appear in the same article and when two entities appear in the same sentence. So when two entities are “closer” inside the text their edge weight increases. Figure 4 demonstrates the knowledge graph created using the text from Figures 2 and 3. The sum of such graphs from all articles of a given time period forms the overall knowledge graph, according to:

$$W(i, j) = \sum_{d \in D} w_d(i, j)$$

(3)

where $D$ is the set of all documents.

### 3.3 Graph Nodes Time Series Analysis

In the context of this paper we explore detection of news events using as an indicator the significant increase or decrease of the weighted degree of an entity node. The All_The_News data set is discretised into 51 one-week blocks spanning 2016. Knowledge graphs are created for all weeks over the year. Then weighted node degree time series are created for all entities detected during the year. By monitoring the evolution of the weighted degree of each node over time, our intention is to detect events based on changes in network structure, revealed in entity time series. To detect large changes we calculate the first differences of the time
series to remove trends, then we calculate the mean and standard deviation from a sliding window of $X$ weeks. An “event” is then indicated by a first difference value exceeding a threshold of $Y$ standard deviations away from the mean. For the example study given here, we set $X = 5$ weeks and $Y$ standard deviations. Depending on the scenario, we can vary $X$ and $Y \in \{5, 7\}$ to consider larger/smaller events or longer/shorter timescales.

3.4 Summarizing the Detected Events

After identifying all peaking entities for each week, event characterization begins by collecting the set of articles for each week that mention the peaking entities. This filter for articles including peaking entities focuses the analysis on entities where some temporal change has occurred; they have become more or less prominent in news coverage, compared to their normal level. Each week group contains documents that describe one or more news events. To help distinguish individual events in the same week we next extract noun phrases from these documents and create a second generation of graphs known as KeyGraphs [Sayyadi et al., 2009]. These graphs include entity nodes alongside keyword nodes corresponding to nouns and noun-phrases, with significance scores assigned as before to derive edge weights. For extracting the noun phrases we use the ToPMine algorithm [El-Kishky et al., 2014].

We use the Louvain community detection algorithm [Blondel et al., 2008], which can be used on undirected weighted graphs, to identify events as communities in the Keygraphs. The entities and keywords in each community then form a bag of words summary of the detected event.
4. Results

In this section we demonstrate our approach and evaluate its performance. First we demonstrate system performance using knowledge graphs, created for a 12-month period, with 1-week windows to explore major news events over the whole of 2016. Then we perform a simple evaluation by comparing system outputs to human annotation for knowledge graphs over a 1-week period with 3-day windows.

4.1 Finding events during 2016

We created a sequence of 51 one-week knowledge graphs using all articles of the All The News data set from the year 2016. Figure 5 summarizes the data analysis and knowledge graph over this time period. A total of 83,601 articles were processed, with weekly minimum of 1,278 and a weekly maximum of 2,046. In total, 249,792 different named entities were detected, consisting of 146,132 persons, 34,235 locations and 69,425 organizations. The weekly numbers of articles and entities were consistent over time, with typically an order of magnitude greater number of entities than articles. After filtering articles to retain only those referencing peaking entities, 25,980 articles were retained overall, with a minimum of 72 articles in Week 13 and a maximum of 1,681 articles in Week 39 (the week of the US Presidential election). For peaking entities/articles, the volume relationship is reversed, with typically around an order of magnitude more peaking articles than entities. Knowledge graph statistics (after filtering for peaking entities/articles; Figure 5, right) show that the weekly knowledge graphs typically contained $10^3$-$10^4$ entity nodes and $10^5$-$10^6$ edges.

![Figure 5: Article and Entity Statistics for the year 2016 of the All The News data set (left). The first 5 weeks of the year were not used for peak detection since they are the lead-in period for the sliding window. Graph statistics for peaking entities for the same period of time (right).](image-url)
For illustration we next focus on a case study of three entities. Figures 6, 7 and 8 show time series for the entities “Nigel Farage”, “Donald Trump” and “Pope Francis”. The highest peak for the “Nigel Farage” entity occurs in Week 25 when the “Brexit” referendum took place (the referendum was held on 23rd July 2016). This makes intuitive sense, since Nigel Farage was a prominent actor in the political debate around UK membership of the European Union due to his role as leader of the United Kingdom Independence Party. For the “Donald Trump” entity the highest peak occurs in Week 45 when the US presidential elections took place. Finally the entity “Pope Francis” presents a peak on Week 6 which is the week that Pope Francis visited Mexico and gave a speech about immigration to the US.

Figure 6: Weighted degree time series for Person entity Nigel Farage. The first detected peak in Week 25 is when the “Brexit” referendum took place, also the second highest peak in Week 45 is when the US Presidential elections took place. The plot on the right demonstrates the significant peaks detected using a threshold of standard deviations $Y=7$.

After the creation of the KeyGraphs we are able to summarize the topic of the detected events in Week 25 by examining the keyword communities that were created. In Figure 9 we present the KeyGraph generated from the collated articles including peaking entities (including Nigel Farage) for the Week 25, with labels added based on manual inspection of the communities. Nodes with weighted degree lower than 1 were filtered out, as were communities with less than 10% of the total number of nodes. The “fuzzy” and potentially overlapping community structure is evident from this plot.

Figure 10 shows the top 100 keywords/entities by weighted node degree for the community labelled as Brexit in Figure 9. Inspection shows a clear association of entities and noun phrases with the Brexit topic, with relevant people (Boris Johnson, Nigel Farage, Michael Gove), places (Brussels, United Kingdom, various European countries) and organizations (European Union, UK Independence Party) mentioned alongside key issues (“member states”, “referendum campaign”, “control borders”, “free movement”).
Figure 7: Weighted degree time series for Person entity Donald Trump. The highest peak in Week 45 is when the US presidential elections took place. We can also recognize the trend of the “Donald Trump” entity as we get closer to the date of the election.

Figure 8: Weighted degree time series for Person entity Pope Francis. The highest peak occurs in Week 6 when he visited Mexico and gave a speech about immigration.

4.2 Evaluation against human annotation

To evaluate our method we manually annotated a total of 1,861 articles from a 8-day period around the Brexit referendum in the UK (19th-26th June 2016). For each article we identified the main story or topic that it reported. In total we detected 851 distinct topics. A large majority of topics were reported in only a single article (~700), while a small number were reported in multiple articles. As an example, in Table 1 we present the most frequent topics in the annotated articles for the 24th of June 2016. In total there were 300 articles on that day, within which we annotated 151 different topics. The most popular topic was the referendum deciding whether the United Kingdom should leave or remain in the European Union (Brexit).
Event detection in news streams

Figure 9: KeyGraph created using the peaking documents of the 25th week of 2016. By examining the keywords of each community we are able to identify the topic of each community/event. Number of nodes: 5303, Number of edges: 288500, Average Degree: 108.8, Average Weighted Degree: 8.9, Diameter 3, Density: 0.021, Modularity: 0.422, Avg. Clustering Coefficient: 0.834.

For comparison we also tested a naive approach that used the K-Means algorithm to cluster articles on 24th June based on word frequencies (Tf-Idf vectorization). To determine the number of clusters we ran the algorithm for 15 clusters and merged similar clusters based on a similarity threshold. After running this approach 20 times, the most informative results were produced using 6 clusters (see Table 1). The k-means method for 24th June found clusters representing the major events of Brexit and the Orlando mass shooting. A third document cluster represented a collection of topics related to US politics. A fourth cluster contained articles relating to the topics of Stonewall and transgender military personnel. The remaining two clusters found using k-means could not be related to any specific event and contained documents relating to different topics. In comparison our knowledge graph method was able to detect almost all of the most frequent human-annotated subjects, although it missed a few important topics including the Donald Trump presidential campaign. After examining the detected KeyGraph communities we concluded that the Donald Trump campaign was omitted because most of the related entities were assigned to the “US Presidential election” community. Comparing the events found using the knowledge graph to k-means clustering, the graph-based method finds a larger number of events. The US politics cluster from k-means corresponded to a combination of four graph-based events (US politics, US election, Donald Trump and Hillary Clinton campaigns). Both
the graph-based and k-means methods lumped together (in some cases) topics/events that were separated by human annotation; this raises the issues of granularity and subjectivity in human annotation, as it is hard to know the correct answer.

To look at event detection over multiple days, we created time series of the daily number of articles that were human-annotated as relating to each unique topic. Then significant news events were found as peaks in article frequency within these time series (i.e. counts larger than one standard deviation from the mean over the previous 3 days). Table 2 shows the set of 6 major news events detected in this way. It is interesting that peak detection from human annotation data found so few events, despite the large number of topics identified. Manual inspection (data not shown) showed that the graph-based approach was able to detect all of the human-annotated events, even though some of them were described from a small number of articles (less than ten).

Figure 11 plots the number of events detected by human annotation, knowledge graph analysis and k-means clustering. While we did not perform a comprehensive analysis of the correspondence between the events detected by each method, the comparison suggests that the graph-based method finds a greater number of events. Consideration of the events for 24th June shown in Table 1 suggests that the events detected by the graph-based method are slightly higher resolution than those found by k-means.
<table>
<thead>
<tr>
<th>Top 20 manually detected topics. (Number of articles)</th>
<th>Detected event communities on the peaking articles KeyGraph</th>
<th>Detected event communities using K-Means clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brexit Referendum: 79</td>
<td>Brexit Referendum 25.55%</td>
<td>Brexit referendum: 80</td>
</tr>
<tr>
<td>Orlando mass shooting - gun control laws: 14</td>
<td>Orlando mass shooting - gun control laws 22.39%</td>
<td>Not identified document cluster (a combination of many subjects): 57</td>
</tr>
<tr>
<td>US Presidential election: 8</td>
<td>Stonewall Uprising monument - Transgender personnel in US military 6.06%</td>
<td>Not identified document cluster (a combination of many subjects): 46</td>
</tr>
<tr>
<td>Hillary Clinton scandal: 5</td>
<td>Hillary Clinton scandal 5.42%</td>
<td>Stonewall uprising monument - Transgender personnel in US military: 36</td>
</tr>
<tr>
<td>US politics: 4</td>
<td>US politics 5.29%</td>
<td>Orlando mass shooting - gun control: 33</td>
</tr>
<tr>
<td>Mike Flynn death: 4</td>
<td>International News 3.97%</td>
<td>-</td>
</tr>
<tr>
<td>Stonewall Uprising monument: 4</td>
<td>West Virginia floods 3.8%</td>
<td>-</td>
</tr>
<tr>
<td>West Virginia floods: 4</td>
<td>Brazil Olympics 3.58%</td>
<td>-</td>
</tr>
<tr>
<td>California wildfires: 3</td>
<td>UK politics 3.5%</td>
<td>-</td>
</tr>
<tr>
<td>Freddy Grey death: 3</td>
<td>North Korea nuclear threat 3.11%</td>
<td>-</td>
</tr>
<tr>
<td>Barack Obama executive: amnesty 3</td>
<td>Ralph Stanley death 2.3%</td>
<td>-</td>
</tr>
<tr>
<td>Stock markets: 3</td>
<td>Stock markets 1.02%</td>
<td>-</td>
</tr>
<tr>
<td>Freddy Grey death: 3</td>
<td>Tennis News 0.51%</td>
<td>-</td>
</tr>
<tr>
<td>EgyptAir crash: 2</td>
<td>EgyptAir crash 0.51%</td>
<td>-</td>
</tr>
<tr>
<td>Hillary Clinton campaign: 2</td>
<td>Dance event Radio City 0.21%</td>
<td>-</td>
</tr>
<tr>
<td>North Korea nuclear threat: 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ralph Stanley death: 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Transgender personnel in US military: 2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Brazil Olympics: 2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Top 20 manually annotated subjects from a single day (24/6/2016, left column) and node percentages of the communities of the KeyGraph created using peaking articles (middle column) and the document clusters formed from a naive K-Means approach (right column).
Table 2: Frequency time series from peaking events of the manually annotated articles. Bold text indicates the day on which there is a significant increase on the topic frequency.

<table>
<thead>
<tr>
<th>Detected Events</th>
<th>19/6</th>
<th>20/6</th>
<th>21/6</th>
<th>22/6</th>
<th>23/6</th>
<th>24/6</th>
<th>25/6</th>
<th>26/6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brexit referendum</td>
<td>11</td>
<td>17</td>
<td>11</td>
<td>18</td>
<td>15</td>
<td><strong>79</strong></td>
<td>37</td>
<td>10</td>
</tr>
<tr>
<td>US election</td>
<td>1</td>
<td>4</td>
<td>18</td>
<td><strong>31</strong></td>
<td>20</td>
<td>8</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Freddie Gray trial</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td><strong>6</strong></td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Marco Rubio election</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Barack Obama executive amnesty</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td><strong>17</strong></td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>George Will left republicans</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>

Figure 11: Number of events detected by human annotation, knowledge graph analysis and k-means clustering. The first three values of the human annotated events are the initial values of the 3-day window.

5. Conclusion

This paper presents a prototype real time news event detection methodology, based on natural language processing and network analysis. Events are located by finding peaks in node-level time series and characterized by community detection in KeyGraphs link-
Event detection in news streams

Ining entities and noun-phrases. The results presented here are preliminary but suggest the method may be effective when suitably developed. Processing information from heterogeneous sources (e.g. Twitter, Reddit, blogs) will make our approach able to detect events on hourly or even smaller time windows. A limitation of the work presented here is the focus only on node degree as an indicator of network change. This results in a large number of candidate events being identified, which require further processing. Filtering the document corpus down to only those articles that mention the peaking entities reduces this problem. The use of network metrics will help to provide a holistic view of the unfolding news narrative and enable the main trends and dynamic patterns to be discerned. Network analysis also has a key role to play in event characterization; we plan to extend and improve the KeyGraph method demonstrated here.

After evaluating our results against the events detected from the manually annotated articles we noticed that our method detects way more events. The reason is that our method monitors all the named entities on each document and detects peaks on their popularity. Manual annotation of the articles has a big disadvantage which is the monitoring of individual entities inside the text. Annotating the behavior of each entity is a very difficult task if not impossible, due to the large number of entities. Our method is able to detect important changes in the popularity of entities and can detect events that are difficult to identify by hand. For that reason our method is able to detect events that are not easily identified by a human.

Finally it is worth mentioning that the number of days used to form the knowledge graph corresponds with the duration of an event. For example events that their date of occurrence is already known (sport events, elections) tend to be mentioned gradually on the news articles and for that reason week knowledge graphs can detect those events more accurately. On the other hand significant events that are not expected (terrorist attacks, accidents) are more accurately detected with day knowledge graphs. For that reason one can adjust the window size depending on what kind of event he is looking for.

In future work we are also going to investigate if we can further split the detected communities into sub communities based on their modularity (or other metrics), so we will be able to form different communities which have many common nodes but different semantic meanings, and if we can assign nodes to more than one communities in order to get a better representation of the event that each community describes.

Acknowledgments

Omitted for blind review.

References


J. Allan and V. Lavrenko. On-line new event detection and tracking. In SIGIR98: Proceedings of the 21st Annual International ACM SIGIR conference on Research and develop-
ment in information retrieval, 1998.


