UNIVERSITEIT VAN AMSTERDAM

Multi-Source Trendwatching

Contextual Analysis of Twitter and Dutch News Websites

HARRIE OOSTERHUIS 10196129
LOTTE WEERTS 10423303

Company supervisor: ROBRECHT JURRIAANS
University supervisor: RAQUEL FERNANDEZ

Informatics Institute, Faculty of Science, Universiteit van Amsterdam

Abstract

Currently available trend watchers, such as the trends-list on Twitter, are capable of supplying a user of a list of frequently used terms. Such trend watchers are limited because the relationship between trending terms and related terms are not presented, resulting in a minimal understanding why a term is trending. Elvers et al. [2011] proposed a trend watcher that is capable of creating networks between trending terms and related terms on textual YouTube content. In this paper we have applied their method to Twitter data. A disadvantage is that the resulting trends are based one resource solely. We propose an extension that uses the TF-IDF ratings of words in data from news websites to improve the relevance of the trends. We found that while the trend watcher as described in Elvers et al. [2011] was capable of detecting relevant trends, over 62% of the found trends could not be defined. The addition of the news filter resulted in an overall decrease of 3,6% of unidentifiable trends and increased the amount of news related trends from 8,3% to 13,7%. However, it also indiced a delay in detecting some news events. These results show that the method of Reed et al. can be used for detecting trends in Twitter. Also, if a bias towards news related content is preferred over the prematurity of the trends, the proposed news filter can improve results.

This paper was written as part of the honours project 2013-2014 of the Bachelor Artificial Intelligence at the University of Amsterdam

Contents

1 Introduction

In this paper a tool is described for recognizing and combining trending topics from several online resources. These topics are represented by a network of words, giving a complete view of the words and their relationships. Twitter data was used to detect trending topics and these were improved using news items retrieved from a selection of Dutch popular news websites. The trend watcher provides the user of a quick view on trending topics and has a potential of processing data near real-time.

Popular online resources such as Twitter already provide a list of what they call *trends*. These trends are merely a list of keywords that are often used at one particular moment. For example, on the fifth of December there is an annual celebration called 'Sinterklaas', a holiday figure based on Saint Nicholas, in which gifts are given to children. Towards the end of November it would not be unlikely for the word *zwarte piet*, the companion of Saint Nicholas, to be trending, since that word is probably used more frequently at that time. The downside of this approach is that it is not easy for the viewer to identify what the exact topic is that is trending based on one key word. For example, recently there was a discussion on the racist background of zwarte piet, which could also make it appear in the trendlists. If *racistisch* (meaning: racist) would not be trending, a user could think it is just because of the time of the year that *zwarte piet* is trending. However, a network of terms that also includes terms that are related but not trending themselves, could include the word *racist* and could make it more evident that that the discussion on the discriminating background of zwarte piet is the actual trending topic. To gain such insight into the reason why a term is trending, we will focus on the relations between the terms that appear to be trending by following a approach as presented by Elvers et al. [2011]. Another disadvantage of the trends that are defined by Twitter itself is that they are only based on one resource. There are a lot of other resources one could think of that could also be used to find trends, for example YouTube, news websites and the subtitles of TV shows. Combining these resources might lead to more comprehensive networks. Using multiple resources comes with difficulties. For example, news items are usually longer than Tweets, but there are substantially less news items available (we could retrieve about 10 news items per hour, but at peak times over 2000 Tweets per minute). Therefore, despite the fact that all sources are text-based, they cannot be used in exactly the same manner. A solution is proposed that uses the TF-IDF value of words in the news item data as an

extra threshold for the trending terms found in the Twitter data.

We will describe software that is capable of creating a network of currently trending topics, based on information retrieved from several online sources. These online sources include Twitter and Dutch news websites. The part of the trend watcher that detects trends in Twitter data is largely based on the work of Elvers et al. [2011]. Our source data will be limited to Dutch resources but should be applicable on various text sources independently of language, provided that the resource is text-based, dividable in documents per user and is not too monotonous (since it must contain trends).

2 Related Work

Trend watchers that can both detect trending terms and also are capable of dealing with multiple sources have to rely strongly on two fields: topic mining and multimedia integration. These two research areas will now be briefly explained.

2.1 Topic mining

Topic mining is an effort in discovering what topics are trending on User Generated Content services (UGC), such as Facebook and Twitter. A system that can detect trending topics in these media allows companies to understand what captures the interest of their users the most. Twitter is perhaps the service that is mostly used for topic mining. A machine learning approach can be useful in detecting trends in Twitter. For example, Lee et al. [2011] used a bag-of-words classification approach to classify tweets in 18 general categories of topics. This method showed to have an accuracy between 65% and 70%. However, in this paper we are not dealing with a classification problem but with an unsupervised learning problem, so other methods should be used. Research on what has triggered a trend has also been conducted. For example, Brennan and Greenstadt [2011] describe a method of real-time trend watching using Naïve Bayes. They recommend using profile information as a feature, because it improved their results. However, using such information results in a method that is service specific. When different sources must be combined, a method that is independent of the source might be preferred. A non service specific method

of topic mining has been proposed by Elvers et al. [2011]. It describes a system that can find emerging trends in any service of User Generated Content that contains text. This system is expanded by adding the concepts nutrition and energy, who capture the relevance and emergence of a topic. A case study performed with the system showed how the 2011 earthquake in Japan could be found present in UGC on the YouTube service, which includes video descriptions and labels. This is remarkable because YouTube is not a service intended to spread news. Because of the non specific nature of their approach, it can be easily applied to different resources. Therefore, we have recreated the proposed implementation of Elvers et al. [2011] and test it by applying to to data from Twitter and news websites.

2.2 Multimedia integration

The second challenge in creating the trend watcher entails the integration of different media sources. Not much research has been done in this area. Roy et al. [2012] present an example of a framework, called *SocialTransfer*, that improves the results of video recommendation by using sources from Twitter. Their project is based on an approach known as Cross-Domain Transfer Learning, where knowledge from one source is used to improve the knowledge on a dataset in a different domain. Their approach consists of two key elements:

1. A technique to determine the topic space
2. A technique to update the topic space real time

Implementing state of the art transfer learning algorithms is out of the scope of this project. However, we will propose another method that integrates data from news websites in order to improve the trend watcher that generally follows these two key steps. Firstly, we will create a topic space using the method of Elvers et al. [2011]. Secondly, we will update this topic space using the TF-IDF of words found in data of news websites.

3 Theory and Method

3.1 Retrieving the data

Firstly, the data from the online resources was retrieved. In order to be able to compare different approaches we simulated a real-time datastream using data that was retrieved in advance. The news items were scraped from ten popular Dutch news sites by downloading the RSS feeds of these websites on a two-minute basis. Additionally, the Twitter streaming API was used to retrieve Dutch Tweets. There are many more Dutch Tweets per two minutes than the ones retrieved, but the Twitter API limits streaming in two ways [1]. Firstly, one needs to determine a list of keywords - retrieving 'random' Tweets is not supported. To prevent the Tweet database from having a bias toward certain trends because of the set of keywords, we used a fraction of the mostly used words in Dutch language [2] to fill this list of keywords. Secondly, the Twitter API limits its users to retrieve only a small fraction of the total volume of Tweets at any given moment, which resulted in a maximum of approximately a thousand Tweets per minute. The code for the scraper was written in JavaScript and uses the Node JS framework. After running it for about a week, approximately ten million Tweets and three thousand news items were retrieved. For both sources not only the post itself, but also the time and date it was posted and the user that posted it were stored. In the case of the news items the user is equal to the name of the website.

3.2 The application

In order to give a clear view on the method used, we take off with a few definitions:

Definitions

Emerging Term A single keyword that is used more than expected

Trending Term An emerging term that is used more than the average emerging term

[1] For more information, see https://dev.twitter.com/docs/streaming-apis
[2] Retrieved from https://onzetaal.nl/taaladvies/advies/woordfrequentie

Contextual Term A word that is often used in combination with a trending term in a strongly-connected way

Trending Topic A network of multiple trending terms and associated contextual terms

After retrieving the data, the following steps were performed on the Twitter data. These steps are, unless stated differently, similar to the approach of Elvers et al. [2011], which was presented as a general way of detecting trends in any given UGC data set. Since news items are of a different nature than User Generated Content the system is not applicable.

1. **Represent each post j as a vector of terms, v_j of length N_t in which N_t is the number of terms in period I**

   In order to efficiently represent the term vectors, a hash table was used. Each word is stored in a static library accompanied by a hash key. For each word in vector v_j the table contains two components: the hash key as stored in the word library and the number of times the term occurs in the post. However, words with a zero number of occurrences are not stored. This reduces the size of the table greatly. Words that have not been stored are assumed to have the value 0.

2. **Weight each entry in v_j by the max term frequency in the post multiplied by $\frac{1}{N_u^I}$, where N_u^I is the number of posts made by the posting user, u, in I.**

   According to Elvers et al. [2011], each term must be weighted with the following formula:

   $$\frac{tf(p_j^I, t_x)}{argmax_i tf(p_j^I, t_i) * N_u^I}$$

   In which the numerator represents the term frequency of t_x and the denominator is the most frequently used term in one post, p_j^I, multiplied by the number of posts the posting user, u, made in I. This last multiplication is needed to prevent that many posts from a single user on one particular topic, which occurs with spam bots, induce a trend.

3. **Sum the weighted term vectors in I**

   The sum of the weights of each term are called the *nutrition*, which is a biological metaphor that represents the importance of a term over a given time interval. Because of the multiplication with $\frac{1}{N_u^I}$, each user can only contribute up to 1 in nutrition. Thus the nutrition of a term is based on the number of users that have used this term, diminishing the effect of their individual activity.

4. **Assign a rank to each term in I, where a rank of 1 is given to the term with the largest combined weight**

   Ranks are used instead of actual weights for they are not affected by the total activity of their particular time frame. For instance, the total activity of a frame that spans nightly hours naturally contains less activity than one that spans an afternoon, thus weights are expected to increase greatly during the morning. To avoid variances like these to effect the emergence of trends, ranks are used instead. Ranks order the terms in relation to each other, making them a better indication of importance.

5. **Model the emergence of each term in I by performing a weighted linear regression using the rank of each term in the previous s time periods and then calculate the fraction of error between the predicted (P) and actual (A) rank value in I via (P - A)/P**

   We decided to use simple linear regression to predict the expected rank value of a term. In general, linear regression can be modeled as solving a least square fit problem, which equals (Bretscher [2011]):

   $$\beta = (X^T X)^{-1} X^T y$$

   In which X is a matrix with the features and y a vector containing the accompanied target value. The advantage of simple linear regression compared to this general approach is that it can be solved without using matrices and therefore avoids the need of calculating a inverse, which is a costly computation. The weights were assigned using a linearly ascending value from 0 to 1 that was multiplied with each tuple. Other forms of linear regression, using a second and third order degree polynomial, were tried but did note significantly improve the results. Therefore, weighted simple linear regression was used. The fraction

of error between the predicted and actual rank, which is called the *energy* of a term, can now be calculated in the following way:

$$energy^I_{t_x} = \frac{P^I(s, t_x) - A^I(t_x)}{P^I(s, t_x)}$$

Where $P^I(s, t_x)$ is the predicted ranking of t_x in I, s is the number of previous intervals that are taken into account and $A^I(t_x)$ is the actual ranking of t_x.

6. **Terms with the fraction of error close to 1 are considered emerging terms for the time period I**

   To find the terms that are to be considered emergent all non-positive terms are disregarded. From the remaining terms the mean and standard deviation are computed. All terms that have an energy greater than two standard deviations + the mean are considered emergent terms.

7. **Create a navigable directed graph, where terms represent nodes and weighted links represent semantic relationships between term pairs**

   To find the emergent terms that are part of the same topic a metric of semantic correlation is used. Unlike traditional correlations that represent a relation between a query and a document, a representation of the directed relation between two terms is used. The correlation, $c^I_{k,z}$, from term k to z:

$$c^I_{k,z} = log(\frac{(n_{k,z} + n_k/N)/(n_z - n_{k,z}) + 1}{(n_k - n_{k,z} + n_k/N)/(N - n_k - n_z + n_{k,z} + 1)})$$

   where:

   - n_k is the number op posts that contain k in I
   - n_z is the number op posts that contain z in I
   - $n_{k,z}$ is the number op posts that contain k and z in I
   - N is the number of total posts in I

   This measures functions well in media where posts are short and are expected to cover a single topic. Because of the short and discrete nature of Twitter posts, this metric seems appropriate.

8. **Extract emerging topics by locating strongly connected components in the graph such that all of the edge weights are above a given threshold and the graph contains at least one emerging term**

   A directed, edge-weighted graph is created containing the emergent terms and all first and second order co-occurring terms. A first-order co-occurring term appears in the same post as an emerging terms, and a second-order co-occurring terms appears in the same post as one of the first-order co-occurring terms. By doing so the graph is kept rather small in comparison to a graph containing all terms in I. The limited size of the graph allows for much faster computation and consumes less memory. The relations between terms are computed using the previously described correlation metric.

   From the resulting graph strongly connected components were extracted using Tarjan's algorithm (Tarjan [1972]) for various threshold values of the correlation weights. Here the threshold values are chosen iteratively so that each emergent term is part of at least one strongly connected component. For each term the component with the highest average energy is taken and components that consist of a single term are discarded. By doing so the resulting components are kept strongly connected yet as small as possible and their size is dependent of how rapidly the threshold decreases each iteration. The resulting components are considered trending topics. We used the D3 JavaScript library to visualize these trends.

3.3 Combining Tweets and news using TF-IDF

One approach to integrate tweets and news items could be to apply the previously described algorithm to news items and combine the two. However, there is a significant difference (about a factor of 300 in our case) in the volume of Twitter data and news items that can be retrieved. Consequently a larger time span is needed to find meaningful trends in news items than than to do with Twitter data (one day and one hour respectively). Daily trends cover profoundly different topics than do hourly trends, so combining those two is not readily feasible. However, there are more ways in which information of news items might be used, for example as a filter. We found that sometimes the results of trends in Twitter data were spoiled by retweets of insignificant messages. For example, there are Twitter users that retweet

messages from companies in order to win a price. Because it is not known if a retweet contains valuable information (such as a news article) or should be regarded as spam, simply removing all retweets from the input did not improve the results. Information retrieved from news articles might be useful for this, since the retweets that contain valuable information might have some familiarity with news articles.

There are several places within the pipeline in which information of news item information might be applied as a filter. One approach could be to filter retweets directly from the input flow, immediately throwing away retweets that do not match enough with the news articles. However, we do not want to remove such terms too early in the process, since this might result in missing interesting trends. For example, words such as *zwartep* turned out to be emerging. This is an abbreviation of the word *zwarte piet*, the helper of Sinterklaas that became controversial because of his possibly racist nature. It might be preferred to remove such words from the final trending topics, since a correct orthography is better understandable. However, by not removing *zwartep* in an earlier stage, it can still be used as an indicator for accompanying words, such as *Sinterklaas*, which will like reoccur with the word *zwartep*. Therefore, we decided to add the filter after the determination of the emergent terms.

In our approach, the energy of each emergent term was temporarily enlarged with a factor according to it's relevance in relation to the retrieved news items. There are several ways thinkable in which the terms used in news posts could contribute to the energy of a term. One could simply use the term frequency, but this would give stop words such as *de* (meaning: *the*), which naturally occur more often, an advantage. To account for this we decided to use the Term Frequency - Inversed Document Frequency or *TF-IDF*. As the name suggests, the TF-IDF of a word is the term frequency, the total frequency a word is used in all documents, multiplied by the inverse document frequency, a measure of how rare a term is across all the documents. It is calculated as follows (Christopher D. Manning and Schtze [2008]):

$$TF * IDF = \sum_{d \in D} t \in d * log(\frac{|D|}{d \in D : t \in d})$$

In which TF is equal to the raw frequency of a term and the IDF is equal to the logarithm of the total number of documents divided by the number of documents the word occurred. By trial and error we found the following formula for calculating the

addition to the energy of each term i:

$$addition_i = 3 * \frac{TFIDF_i}{TFIDF_{max}} * variance$$

In which *variance* equals the variance of the emergent terms, which was also used in calculating the threshold, and $TFIDF_{max}$ is the maximal TFIDF for this set of emergent terms. The energy of each term is temporarily enlarged with this addition. The threshold of *mean + 2*variance* was enlarged to *mean + 2.2*variance*, based on empirical research. By doing so, terms with some relevance to the news items are preferred, but terms that are really emergent ($>$ mean + 2.2*variance) but not related to the news items are still retained. Note that afterwards (in creating the trending topics) the original energy is used.

4 Experiments

4.1 Results of trend watching Twitter

To examine the effectiveness of our system we have gathered Twitter posts between 1/11/2013 and 19/11/2013. Each day around 1.5 million Twitter posts were collected.

4.1.1 Setting the parameters

First, the number of time frames used for the prediction of the word ranking needs to be set. Because of memory constraints, we decided to use five time frames. This allowed us to run simulations without a dedicated server, greatly increasing the speed of the analysis. An important aspect of getting the wanted results is determining the proper time interval. Elvers et al. [2011] they used a time frame of 24 hours for YouTube content. A smaller interval allows the system to recognize more trends and to do so faster. In this paper we consider news events the most important trends and found that intervals of an hour or greater are best to be used. Smaller intervals found too many trivial trends such as topics regarding fast-food discounts. Although this may be very interesting for that particular industry, we decided using intervals of an hour allowing the system to recognize topics early while still preventing trivial

topics from emerging. Figure 1 shows the percentage of trending topics per time span relative to the largest amount of trending topics of that particular time span. We found that the meaningfulness of the topics somewhat correlated with this relative amount of trending topics. The three hour time span follows generally the same pattern as the one hour time span and we found that the detected trends were of a comparable meaningfulness. The six hour time span shows a valley at 12:00, which reflects a large lag relative to the other two time spans. We also found that the detected trends were not very meaningful. Apparently, interesting trends are not kept alive for longer than three hours on Twitter. Because of the large amount of memory and computation time needed to calculate these trends, we decided to use the one hour time span.

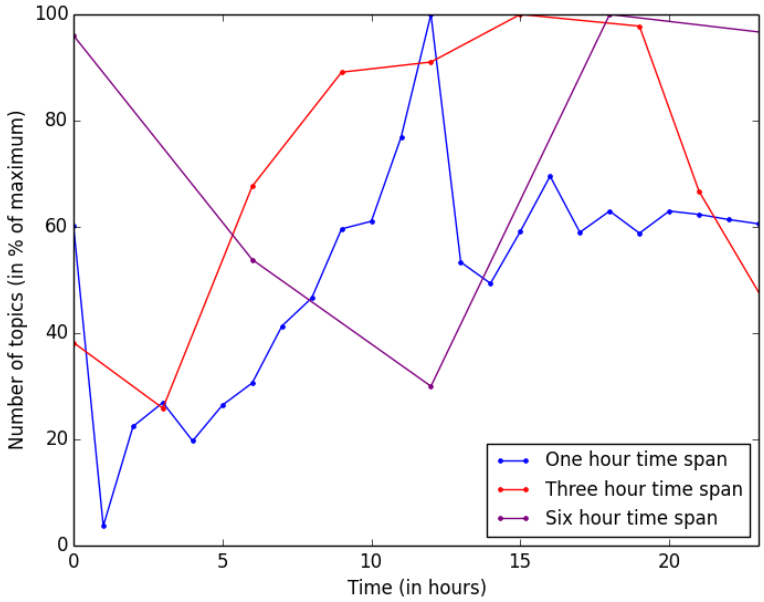

Figure 1: Number of topics as a percentage of the maximum number of topics over one day per time span

4.1.2 Qualitative analysis of results

The system recognized a large amount of topics, the majority covering events that we would deem unimportant. A shown in the left column of table 1, about 62% was categorized as 'not defined', meaning those trends were either nonsense or not recognized as one of the other categories. This is comes as no surprise since one can not expect something important to happen every hour of the day. To illustrate the

strengths and weaknesses we have sampled three topics for discussion. The first two, displayed in figure 2, show trending components regarding irrelevant trends.

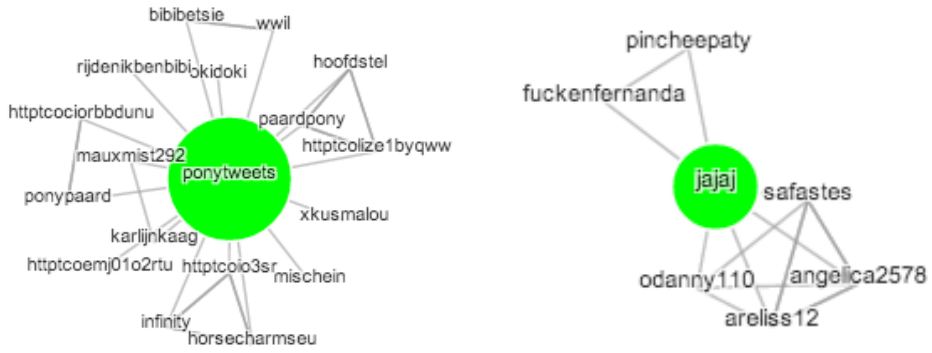

Figure 2: One trending topic regarding a popular pony related accounts, the other regarding an online joke that has become an ”accidental” trending topic.

The first is caused by a twitter account dedicated to children who like ponies, during that time frame one of their posts proved to be very popular. The same effect can be witnessed when other popular content is published for instance when a popular YouTube channel publishes a new video. The second component is related to an online joke picture posted by a Spanish user, we consider this topic to be ”accidentally” trending. The only trending term inside the topic is ”jajaj” indicating laughter in Spanish (as the English ”haha”). The reason this term is considered emerging comes from the fact that it is Spanish and it is misspelled. The misspelling makes it usage unexpected, as opposed to ”jaja” which is more commonly used and the rarity in which is used in the Dutch language. These features give the term ”jajaj” a much higher energy than ”haha” or ”jaja”, even though they refer to the same thing.

On 17 November 19:20 local time 50 people on board a Boeing 737 from Moscow Domodedovo airport died in explosion on runway at Kazan airport. An hour later the system recognized this topic as the number two highest trending topic, as depicted in figure 3. An hour later the topic appeared again, this time consisting of Dutch terms. The terms in the topic that are not hyperlinks are: *plane, crash, aircrash, dozens, sonkoerant, russian, gesen, b737, boland*. Besides the accident, the nationality and flight number of the flight have been identified. Remarkably the system recognizes this several minutes before it is reported on the Dutch news websites. It is worth mentioning that the three hour and six hour time span did not detect the plane crash as trending topic.

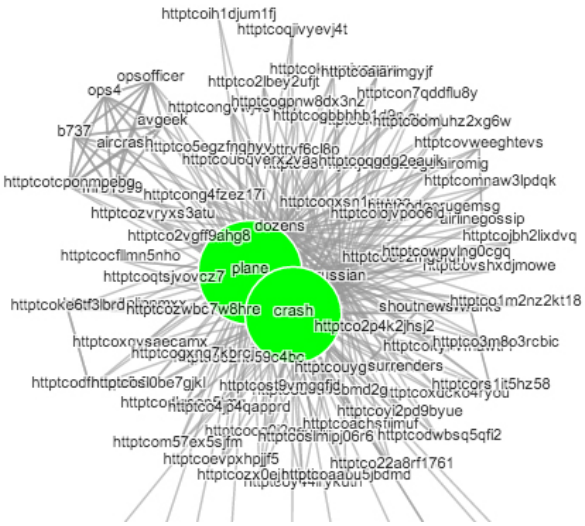

Figure 3: Trending topic on 17 November regarding a plane accident in Russia.

4.2 Effects of the news filter

4.2.1 Quantitative analysis

A comparison was made between the top ten topics created in a one hour time span with and without news filter, using data of the 17th of November, from 05:00 to 19:00. The time span of the news items used for calculating the TF-IDF was four hours before each Twitter time frame, based on empirical experimentation. The news websites that were used in retrieving the data were the following: *volkskrant.nl*, *nu.nl*, *nrc.nl*, *ad.nl*, *trouw.nl*, *telegraaf.nl*, *spitsnieuws.nl* and *metronieuws.nl*. An overview of the found developments are shown in table 1. The table shows that the addition of the news filter leads to an additional 4,67% of news item related content. Additionally, the amount of tv-show related content is doubled. The number of trending artists (such as musicians), recreational events (such as a pop concert) and inappropriate content (such as sexually explicit advertisement) are outweighed. Twitter related content, which includes popular hashtags such as '#ikwilvoor1dag' and popular Twitter users, reduces with -0,67%. Both trends on football and popular brands decrease, 1,43% and 2,00% respectively. Overall, the addition of the news filter seems to result in a higher relevance, since the 'undefined' category reduces with 3,57%. Note that it might be possible that some of the trends in this category were notable trends that were simply not recognized.

Category	No filter (%)	News filter (%)	Relative difference (%)
News item	8,33	13,67	4,67
TV shows	1,33	2,67	1,33
Artists	5,33	5,33	0,00
Recreational event	2,00	2,00	0,00
Twitter related	10,33	9,67	-0,67
Inappropriate content	1,33	1,33	0,00
Brands	3,33	1,33	-2,00
Football	4,67	2,67	-1,33
Undefined	62,14	58,57	-3,57

Table 1: Effects of the news filter on November the 17th from 6:00 to 19:00

4.2.2 Qualitative analysis

Due to the subjective nature of evaluating an unsupervised learning problem, the effects of the news filter may be best explained with a qualitative analysis. For example, at 05:00 the trend watcher with news filter detected the trend *China* and *elf* (meaning: eleven). The trend, depicted in figure 4, refers to an attack on a police station in China, where eleven people were killed. This news item does not show up at the trend watcher without filter.

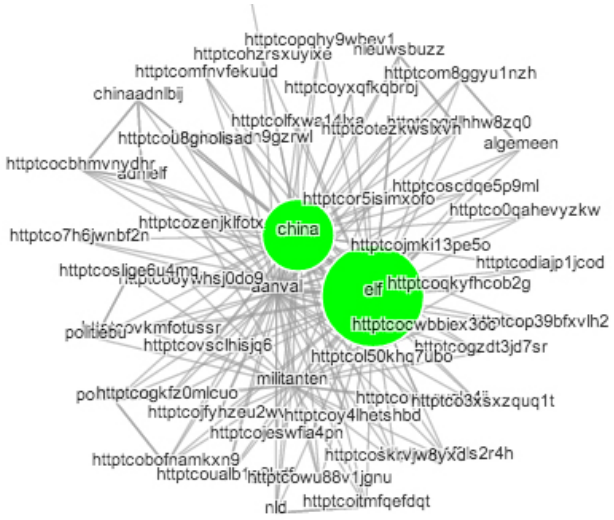

Figure 4: The eleven deaths due to an attack on a police station in China shows up at 05:00 in the trend watcher with news filter

Another noticeable example is the one at 07:00, where both versions of the trend watcher noticed that the event Glow, a festival in the city centre of Eindhoven dedic-

ated to light, received a record number of visitors. As shown in figure 5, whereas the trend watcher without filter only picks up the trending term *recordaantal* (meaning: record number) the trend watcher with filter also appoints *glow* and *lichtfestival* (meaning: light festival) as trending terms, giving a more complete image of the trending topic.

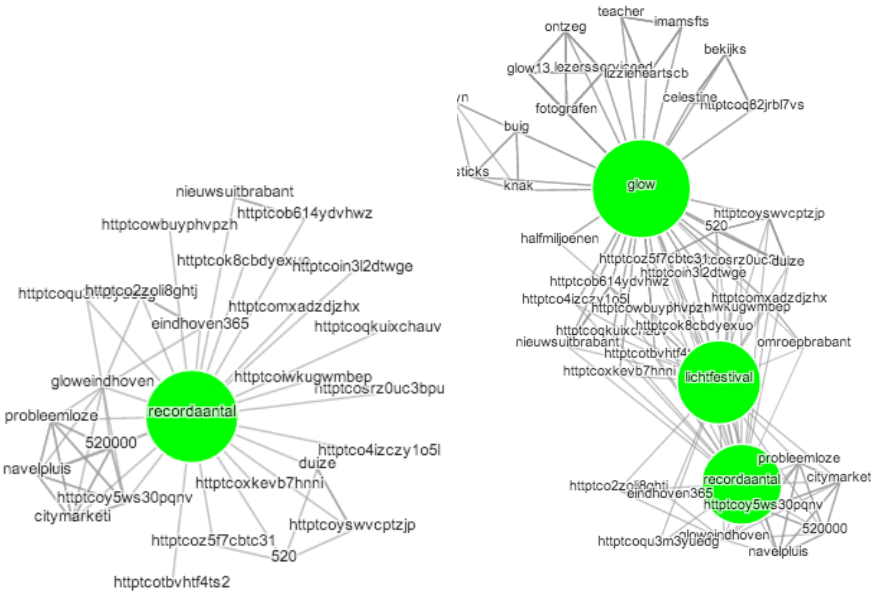

Figure 5: The glow-trend without news filter (left) and with news filter (right)

Figure 6: The plane crash at 18:00 with news filter

Another example pops up at 17:00 and 18:00. The previously discussed plane crash

was already visible at 17:00 in the trend watcher without news filter. The addition of the news filter added an hour delay, making the plane crash pop up at 18:00, as depicted in figure 6. However, the topic created with news filter at 18:00 seems to include different words and with less links. For example, words such as *moskou*, *boeing*, *ministry* and *neerges* (probabily an abbreviation of *neergestort*, meaning crashed) were not visible in the trend predictions without filter. Additionally, the Dutch word for plane crash (*vliegtuigcrash*) is only a trending term in the trend watcher without news filter.

5 Conclusion

In this paper we applied the user generated content emerging topic detection as described in Elvers et al. [2011] to Twitter. While it was capable of correct detection the system, still 62,1% of the found trends could not be recognized. Nonsense words such as *jajaj* are still present in the results. However, the system was also capable of detecting meaningful trends, such as the air crash on the 17th of November. The system proposed by Reed et al. was successfully extended with information from news websites. By using the TF-IDF value of words in news items the amount of undefined topics was reduced to 58,6%. Moreover, the number of news item related trends increased from 8,3% to 13,7%. This shows that using different sources can be used to reduce the amount of nonsense trends and increase the amount of trends that are related with the extra source. However, by using news items for filtering news a delay in the trend detection was found. Whereas the air plane crash of the 17th of november was detected even before the Dutch news sites wrote about it when no filter was used, the addition of the news filter resulted in a one hour delay. It depends on the purpose of the trend watcher whether or not the addition of the filter is desirable. For indicating trends as soon as possible, the news filter could better be left out. On the other hand, if the accuracy of the trends is more important, we recommend using the news filter.

6 Future Work

One major flaw in the approach described in this paper is that it assumes the user generated content has a correct orthography. Since misspelled or nonsense words are not used very often over time, it is quite easy to make them exceed the expectation. Especially in Twitter misspelled words tend to be replicated in one time span, due to the retweet function. An extension of the system that recognizes misspelled words and deals with them accordingly could dismiss such 'accidental' trending topics. One approach could be to use a language model that describes the probability distribution of letters in a word based on a corpus that belongs to the particular source, which in our case could be Twitter data. By using a corpus of the same source, source specific characteristics (such as hyperlinks and "RT" in twitter) are taken into account. Words that turn out to be very unlikely to occur could then be removed.

Another possible extension of the current implementation is to use the system for detecting trends in news items and to merge these with the trends detected by Twitter, to retrieve a more complete view on what is trending. As stated in this paper, we found that a time span of a week was not enough to identify valuable topics in news items. Applying the system to a dataset that covers a larger time span might return better results. Another way to improve the results for news items might be to use a different correlation metric that accounts better for the occurrence of multiple topics in a single document. Still, the challenge of combining the trends found in Twitter and in the news items remains.

References

M. Brennan and R. Greenstadt. Coalescing twitter trends: The under-utilization of machine learning in social media. pages 641–646, 2011. doi: 10.1109/PASSAT/SocialCom.2011.160.

O. Bretscher. *Linear algebra with applications*. Pearson, 2011.

Prabhakar Raghavan Christopher D. Manning and Hinrich Schtze. *Scoring, term weighting, and the vector space model*. Cambridge University Press, 2008. URL http://dx.doi.org/10.1017/CBO9780511809071.007.

T. Elvers, C. Todd, and P. Srinivasan. What's trending?: mining topical trends in UGC systems with YouTube as a case study. *Proceedings of the Eleventh International Workshop on Multimedia Data Mining*, 2011.

K. Lee, D. Palsetia, R. Narayanan, M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. pages 251–258, 2011. doi: 10.1109/ICDMW.2011.171. URL http://dx.doi.org/10.1109/ICDMW.2011.171.

S.D. Roy, T. Mei, W. Zeng, and S. Li. Socialtransfer: Cross-domain transfer learning from social streams for media applications. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 649–658, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1089-5. doi: 10.1145/2393347.2393437. URL http://doi.acm.org/10.1145/2393347.2393437.

R. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1972.