Data Harvesting and Event Detection from Czech Twitter

Václav Rajtmajer[1] and Pavel Král[1,2]

[1] Dept. of Computer Science & Engineering
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
[2] NTIS - New Technologies for the Information Society
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
{pkral,rajtmajv}@kiv.zcu.cz
http://nlp.kiv.zcu.cz

Abstract. Twitter belongs to the fastest-growing microblogging and online social media. Automatically monitoring and analyzing this rich and continuous data stream can yield valuable information, which enable users and organizations to discover important knowledge. This paper proposes a method for harvesting of important messages from Czech Twitter with high download speed and an approach to discover automatically the events in such data. We identified important Twitter users and then we use these lists to discover potentially interesting tweets to download. The tweets are then clustered in order to discover the events. Final decision is based on the thresholding. We show that the harvesting method downloads about 6 times more data than the other approaches. We further report promising results of the event detection approach on a small corpus of the Czech Tweets.

Keywords: Czech, Clustering, Data, Event Detection, Harvesting, Social Media, Twitter

1 Introduction

Microblogging is a novel broadcast (social) medium that allows to create, share, view and analyze information particularly in a form of short messages. Although this service is relatively new compared to traditional media, it has gained significant popularity among individuals, companies and other organizations. For example, during recent social upheavals and crises, many people on the planet used Twitter to report and follow the main events. The importance and the size of the today's social media are growing very rapidly which is strictly related to the particular needs of the automatic processing methods.

Twitter is currently the most fastest-growing microblogging medium, with more than 250 million users producing about 500 million tweets[1] per day[2]. The tweets can be accompanied by photos, videos, geolocation, links to other users and trending topics. The posted tweet can be liked, commented by the other tweets, or redistributed by other users by forwarding, so-called *retweet*. Due to its simplicity and easy access, Twitter contains a wide range of topics from common every day conversations over sport news to information about ongoing disasters as earthquake, flood or typhoon.

Twitter is thus certainly an interesting source of real-time information which can be used for further analysis and data-mining. In this work, we use Twitter because of its large size, significant amount of other existing work dealing with this network and particularly because of a number of Twitter users post interesting news from various topics in real-time. We will use Twitter for automatic real-time event detection because it will be very useful for many journals and news agencies in order to discover quickly new interesting information. Particularly, the Czech News Agency (ČTK[3]) needs a system to automatically collect data from Czech Twitter and on-line discover potential events.

Our first task consists in analyzing Twitter stream and harvesting the appropriate Czech tweets in real-time. The second important task is the subsequent analysis of the downloaded data to discover new events. We have described in [8] a novel Twitter harvesting approach with high download speed. However the event detection was not evaluated precisely, because of the lack of an annotated corpus. This paper extend this work and first presents a small Czech Twitter corpus annotated with the events. Then, the event detection is evaluated on this corpus.

The core of the harvesting method relies on using user lists to download a sufficient number of Czech tweets in real-time. The tweets are then clustered to groups in order to discover the events. Final decision is based on the thresholding.

Several definitions of events exist, however we will use in this work the definition from Cambridge dictionary where an event is defined as "anything that happens, especially something important and unusual[4]".

The paper structure is as follows. Section 2 is gives a short review of Twitter analysis with a particular focus on event detection. Section 3 presents the proposed approach for Czech Twitter analysis and event detection. The following section describes the architecture of the proposed event detection system. Section 5 deals with the results of our experiments. In the last section, we conclude the paper and propose some future research directions.

2 Related Work

Twitter offers a number of possibilities for data processing and analysis, it is thus investigated by many recent research works. The data from this social medium can be used for instance for opinion mining or sentiment analysis as shown in [11]. In this paper, a corpus for sentiment analysis on Twitter is presented and an efficient sentiment

[1] short messages limited by 140 characters

[2] http://www.internetlivestats.com/twitter-statistics/ - June 2017

[3] http://www.ctk.eu/

[4] http://dictionary.cambridge.org/dictionary/british/event

classifier is further built on this data. Another work about sentiment analysis on Twitter is proposed in [7]. This paper deals with the importance of linguistic features for this task. The data from Twitter can also be used for sociological surveys as presented in [14]. This paper analyzes a group polarization using the data collected from dynamic debates. Another paper [5] deals with Twitter analysis to discover user activities. The authors present a taxonomy characterizing the underlying user intentions.

Twitter is also successfully used for event detection task as presented for instance in [13,2]. These approaches generally capture a presence or an increase of important key-words. A presence of words “hurricane” or “inundation” is used to detect the catastrophes.

It has been also suggested many sophisticated Twitter analysis methods as for instance in [9]. This paper describes a system called *Twevent*, which first detects the segments of events and then, they are clustered considering both their frequency distribution and content similarity for event detection. The authors use Wikipedia as a knowledge base to identify the most interesting segments to discover realistic events. The main advantage of this method from the previous ones is that it is domain independent and therefore, it can discover all types of the events.

Recently, bursty event detection techniques [4] have been emergent. These methods lie on the assumption that previously unseen or rapidly growing topics in the stream should represent new events. They thus consider an event in data streams as a bursty activity, with several features high rising frequency as the event emerges. An event is therefore represented by a set of keywords showing burst in appearance numbers [6]. The authors assume that some related words would show an increased usage as an event occurs. These techniques analyze distributions of features and detect events by grouping bursty features with identical topics. The further event detection approaches from Twitter are available for instance in the survey [1].

Twitter analysis approaches are concentrated particularly on English (sometimes also on French or on Chinese). Relatively few works are focused on the other languages. Twitter activity of the users in such languages is high and therefore common harvesting methods provided by Twitter API are sufficient to get enough of data for a further analysis. Note, that only few work about automatic event detection from Czech Twitter exists.

3 Czech Twitter Analysis & Event Detection

This section represents the main research contribution of this paper. We describe here three methods dedicated to data acquisition from Twitter with a particular focus on the proposed method based on the list of the users. Then we details the pre-procession and our event detection approach. All these tasks are necessary to build our event detection system.

3.1 Data Acquisition

The data harvesting method must fulfill the following properties:

- harvesting a “sufficient” number of tweets;

- downloading Czech Tweets;
- usage for free;
- working in real-time;
- connection to the Java programming language;
- downloading only informative messages (optional).

We descibe next the different data harvesting possibilities of Twitter. We compare methods provided by Twitter API with the proposed method. We show for all the methods the maximum download speed defined by Twitter protocols. However, this speed usually differ from the real one, because the activity of the users of Twitter is not sufficient to fill these limits.

Twitter API is a publicly available library that serves to access to Twitter data. Although its simplicity, it contains numerous practical functions for user authentication, accounts management, tweet control and so on. The main functionality is provided for free and is available for every registered user. The communication with the API uses HTTP queries and the response containing the data is in JSON[5] format.

Search API Method Search API belongs to the Twitter REST API which provides program access for reading/writing Twitter data. The functions from this library include for instance the creation of a new tweet, reading of user profile, searching a tweet or user, etc.

Search API allows queries against the indices of recent or popular tweets and behaves similarly to, but not exactly like the search feature available in web clients. This API searches against a sampling of "recent" tweets published in the past 7 days and the maximum download speed is 72,000 tweets/hour. It is possible to restrict the query by several constraints as for instance by a geolocation or by a target language.

This API is focused on relevance and not on completeness. This means that some tweets and users may be missing from the search results. The first approach, which is further evaluated and compared, uses this API. It is hereafter called as *Search API* method.

Filtered Streaming API Method Streaming API is used for on-line Twitter monitoring (or processing). It is stated that this API represents the fastest continuous access to tweets.

There are three different streams with three different connection types available. The first one, *User stream*, is dedicated to use for providing of the data of the connected user. The second one is *Site stream*. This is an adapted version of the previous one for more specified users. The third one, *Public stream*, allows to get public data from different users and about different topics. We need to harvest the data from a open set of users, therefore only *Public stream* can be suitable for our task.

Twitter proposes *Sample*, *Filter* and *Firehose* connection types. The *Sample* connection provides only a sample from all the data without a good possibility to manage queries. *Firehose* connection returns all possible data and could be suitable for our task.

[5] JavaScript Object Notation is a lightweight format for data exchange.

However, this option is not free of charge. The last possibility, *Filter* connection, facilitates tweets filtering according to several critters and is free of charge. Therefore, based on the characteristics mentioned above, we can use only *Filter* connection.

The query can be, as in the case of the *Search API*, restricted by several constraints (e.g. geolocation or target language). The maximum download speed of this method is unfortunately not given. The second evaluated approach uses this API and is hereafter referred as *Filtered Streaming API (FSA)* method.

UserList Method UserList is a Twitter functionality to allow each user to create 20 lists with the possibility to save up to 5,000 users into one list. These lists can be used to show all tweets that these users have posted and this procedure can be used with Twitter API to get all published data from 100,000 particular users.

The proposed method uses these list for harvesting of the significant amount of tweets in a given language (in Czech in our case, however the method is sufficiently general to handle the other ones). The downloaded messages should contain valuable information for data-mining and further analysis as for instance potentials events.

Our issue is now to select the representative users in order to detect appropriate tweets. Our system is designed for general event detection. Therefore it must cover the all Twitter topics by active authors from all fields. We use a small sample of interesting people provided by Czech News Agency and this sample is automatically extended by our algorithm.

The proposal of this method is based on the following observations:

- Our preliminary studies have shown that the methods provided by Twitter API are not suitable for our goal;
- about 20% of Twitter users post informative tweets, whereas the remaining 80% not [10];
- we have already a representative group of the users (sample provided by ČTK);
- this set covers a significant part of our domain of interest;
- their followers would be the users with similar interests.

Therefore, we get using the Twitter API detailed information about all the followers of our initial group. Then, we filter out all foreign (no Czech) users and we continue with the first step. When a requested number of the users is explored, the algorithm is stopped.

For every user u, it is then computed a rank R_u which is based on its number of followers Fn and the number of submitted tweets Tn as follows:

$$R_u = w.Fn + (1-w).Tn \tag{1}$$

where w is the importance of both criteria and was set experimentally to 0.5.

Our list is sorted by this rank and the “best” 100,000 users are added to our twitter lists for a further processing. We use Twitter API to save the data from this representative list of the 100,000 users to our data storage.

Twitter social medium is dynamic and it is modified very quickly. Therefore, this list must be periodically updated to keep actual information. It is also worth of noting that this method is language independent. This algorithm is hereafter called *UserList* method.

3.2 Pre-processing

Spam Filtering We realize spam filtering in order to remove non informative useless tweets. These tweets are filtered with a manually defined list of entire tweets or set of rules. Table 1 shows some examples of whole tweets. The rules are based on the predefined patterns.

Table 1. Examples of the tweets to filter

Tweet	**English translation**
Automatically created messages	
Přidal jsem novou zprávu na Facebook.	I have added a new message on Facebook.
Líbí se mi video @YouTube.	I like @YouTube movie.
Označil(-a) jsem video @YouTube.	I have marked @YouTube movie.
(Everyday) useless tweets created by the users	
Dobré odpoledne!	Good afternoon!
Jdu večeřet, dobrou chuť.	I'm going to have dinner, enjoy your meal.

Note that this method cannot guarantee to filter all useless tweets due to its simplicity. However, we assume that they will not be detected as events by our detection algorithm due to their not significant amount. Therefore, it is not necessary to implement more sophisticated filtering approach for the current system.

Lemmatization Lemmatization replaces the particular (inflected) word form by its base form lemma (base form). It thus decreases the number of parameters of the system and is successfully used in many natural language processing and related tasks. We assume that lemmatization can improve the detection performance of our method. It can be useful particularly in clustering to group together appropriate words.

Following the definition from the Prague Dependency Treebank (PDT) 2.0 [15] project, we use only the first part of the lemma. This is a unique identifier of the lexical item (e.g. infinitive for a verb), possibly followed by a digit to disambiguate different lemmas with the same base forms. For instance, the Czech word "třeba", having the identical lemma, can signify *necessary* or *for example* depending on the context. This is in the PDT notation differentiated by two lemmas: "třeba-1" and "treba-2". The second part containing additional information about the lemma, such as semantic or derivational information, is not taken into account in this work.

Note that we also envisaged to use stemming, an analogical process to lemmatization, which also reduces inflected words to their root form as called "stem". The stem need not be identical to the morphological root of the word. It is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. However, based on our previous experiments [3], we decided to use lemmatization, rather than stemming.

Non-Significant Word Filtering Non-significant (or stop) words are words with high frequencies which have in a sentence rather grammatical meaning as for instance prepositions or conjunctions. Our filtering is based on a manually defined list. We will implement more sophisticated method based for instance on Part-of-Speech (POS) tags in the further version. However, we assume that this improving will play only marginal role for event detection.

3.3 Event Detection

One-pass Clustering We propose a one-pass clustering method to extract the events. We consider that we have in real-time the filtered and lemmatized tweets which can represent (due to the UserList method) probably the events. We transform every tweet into a binary representation using a bag of words method, which represents its unique location in n-dimensional space. Then the clustering algorithm is as follows:

1. take an (unprocessed) tweet;
2. calculate the cosine distance between a vector representing this tweet and all the others;
3. choose a closest tweet (or cluster of tweets if any) and group them together (the maximum allowed distance is given by the threshold *Th*;
4. repeat the two previous operations (*go to step 1*) till all tweets are processed.

The clusters created by this algorithm represent the events. Of course, the clustering does not guarantee that the created clusters represent only the events. This should be done by the pre-processing:

- UserList data acquisition method harvests particularly informative tweets which contains mainly the events;
- spam filtering step removes several non informative useless tweets (no events).

We also define a parameter T which indicates a time period for the clustering. We assume that different events will be produced at different “speed” (different activities of Twitter users). For instance, information about the winner of the football championship can be quicker (more contributions in a short period) than information about a new director of some company.

It is worth of noting, that we have also considered a *gradient* of the word frequencies in some event clusters. It means, that a significant increase of the presence of particular word/words indicates probably an event. However, this approach did not work because of the small activity of the users on Czech Twitter.

Results Representation The results of the clustering are thus the groups of tweets with some common words. This group is represented by the *most significant* tweet. This tweet is defined as a message with the maximum of common words and the minimum of the other words. This representation is used due to the effort to use an answer in natural language, instead of a list of key-words or a phrase.

4 System Description

We describe in the following text the main properties and general architecture of the proposed event detection system. The system is implemented under Java 8 programming language and has a modular architecture. It is composed of three main functional units (*Tweet Stream Analysis*, *Preprocessing* and *Event Detection*) which are further decomposed into six modules, as shown in Figure 1.

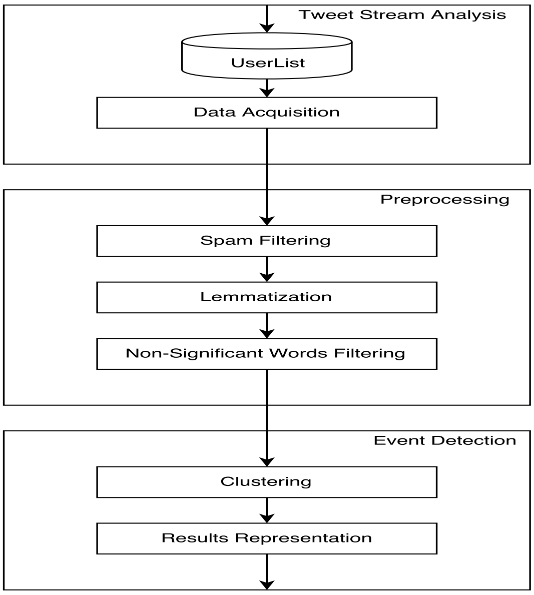


Fig. 1. Architecture of the System [8]

M1 : Data Acquisition The first module, *Data Acquisition*, is used for on-line storage of appropriate information from Twitter stream in Czech language with high download speed. We integrate the proposed UserList method into this module to harvest a sufficient number of tweets. The output of this module are raw tweets without any post-processing.

M2 : Spam Filtering The second one is *Spam filtering* module. It removes tweets with useless information (so called "spam") using rule-based approach. The input are raw tweets collected by the previous module *M1* and the output consists of a set of partially filtered tweets.

M3 : Lemmatization The third, *Lemmatization* module is used for word normalization to decrease the number of feature in the system. The input are partially filtered tweets provided by the module *M2* and this module outputs lemmatized text.

M4 : Non-significant Word Filtering The next one is *Non-significant word filtering* module. While the previous filtering was at the tweet level, this one filters the words and is thus used to remove non-significant words which could decrease the detection performance.

M5 : Clustering Module The *Clustering* module is used to discover the events. We group together the tweets with similar content using a one-pass clustering method. The final decision about detected events is based on thresholding. The input of this module is the text of lemmatized and filtered tweets and this module provides information about the discovered events.

M6 : Results Representation This last module is used to show the detected events to the users in an acceptable form. One tweet representing the detected event is presented to the user in this step.

5 Evaluation

We describe next the experiments realized to evaluate the proposed Twitter harvesting method. This section further details the results of our event detection approach.

5.1 Data Acquisition

Comparison of the Czech and French Twitter Activity In this experiment, we confirm our claim that the activity of the Czech Twitter is significantly lower than in the case of the other languages. We have chosen French Twitter and the *Search API* method (see Section 3.1) for such comparison.

First, we have shown that it is not possible to use language constraints to obtain only the Czech tweets. However, the Czech constraint is missing and there is available only "sk" field which contains Czech and Slovak tweets together.

Therefore, we have decided to filter tweets according to the geolocation. We have chosen a square region, covering most of the territories of the Czech Republic and France, as our area of interest. We have analyzed the download rate in interval from 22 to 29 August 2015. Figure 2 shows the results of this analysis.

This figure shows that the activity of French Twitter is more than 10 × higher than the Czech Twitter. The average of the Czech download rate is about 495 tweets/hour. However, after a detailed examination, we have identified that only less than 20% of tweets are written in Czech languages.

Unfortunately, this number is insufficient for a successful further analysis as for instance for event detection in real-time. Therefore, we must analyze the other approaches for data acquisition.

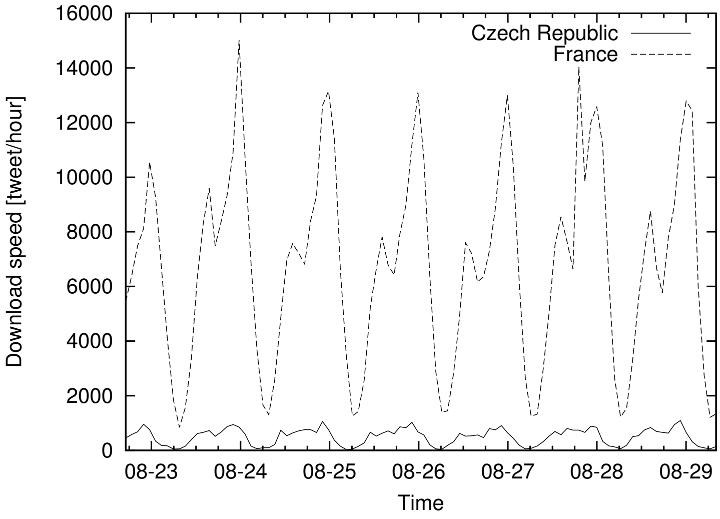


Fig. 2. Comparison of Czech and French Twitter activity [8]

Comparison of the Different Data Acquisition Methods In this experiment, we compare the download speed of two standard methods provided by the Twitter API (namely *Search API* and *Filtered Streaming API (FSA)* methods with the proposed *UserList* approach (see Sec. 3.1). We have thus executed all these methods in the same two day period and then we have calculated the average value for one hour.

Table 2. Comparison of the download speed of the different methods on the Czech Twitter

Method	Search API	FSA	UserList *(proposed)*
Download speed [Tweets / hour]	43.5	56.6	**330.3**

The results of this experiment are shown in Table 2. This table shows that the proposed method provides about 6 times more data than the standard methods provided by Twitter API. Based on these results we have chosen the *UserList* approach to integrate into our event detection system.

5.2 Event Detection

Event Corpus We created a small corpus of Czech tweets to evaluate the event detection method. It is composed of 536 Tweets collected during one hour using the proposed *UserList* data acquisition method described previously. This corpus contains 4 978 word tokens being 3 532 different words and 39 events. It was annotated by two different annotators with annotator agreement 87.9%. The ambiguous cases was decided by the third experienced annotator.

Detection Results Figure 3 shows the results of the proposed event detection method depending on the different acceptation threshold *Th*. The standard recall, precision and f-measure (*F1*) metrics [12] are used for evaluation of this experiment. This figure shows that the best detection results are obtained with the threshold value *Th = 0.6*. This figure further shows that precision values are significantly higher than the values of recall.

The obtained results are promising, although the resulting f-measure is not much high and should be further improved. It could be done by the extension of the corpus and by adaptation of the clustering method. On the larger corpus, it will be possible to use longer time periods (two, four, six hours, etc.) to detect the events that are written at different speeds, which cannot be detected on the current version of the corpus. We also envisage to use word embeddings for better representation of words in the clustering method instead of the simple bag of words. This representation is much more precious and it should create better clusters of the tweets.

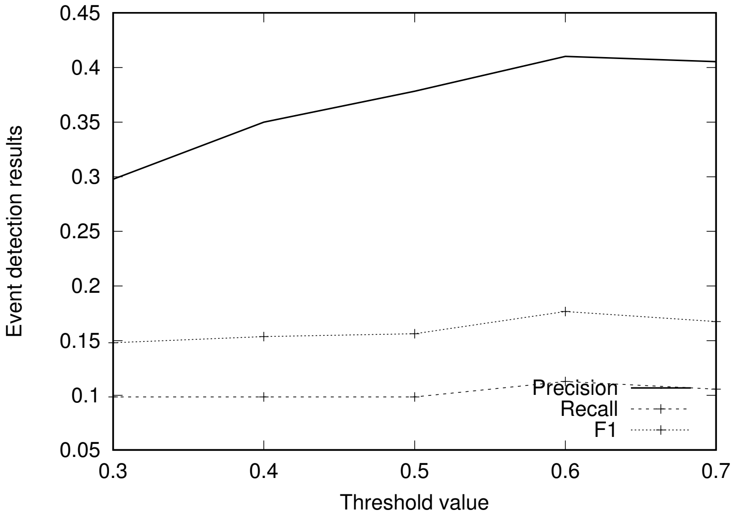
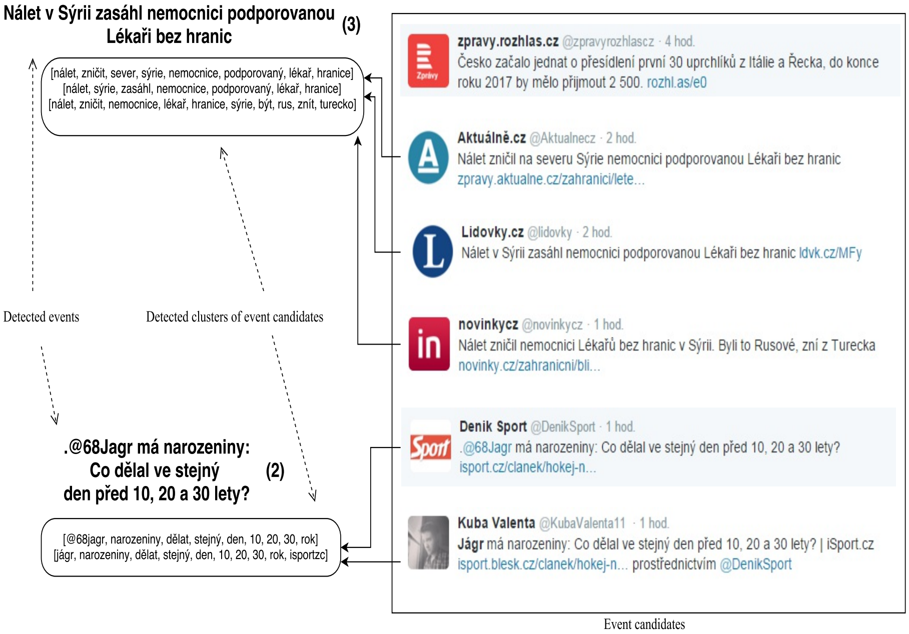


Fig. 3. Event detection results dependent on different acceptance threshold *Th*

Detection Example The sample results are depicted in Figure 4. This figure shows that six tweets are saved by our acquisition method (right rectangle). They are then clustered into two groups containing three and two tweets (left "bubbles"). Finally, one representative tweet is chosen from both clusters to be presented to the user (bold text left).


Nálet v Sýrii zasáhl nemocnici podporovanou Lékaři bez hranic
(3)
[nálet, zničit, sever, sýrie, nemocnice, podporovaný, lékař, hranice]
[nálet, sýrie, zasáhl, nemocnice, podporovaný, lékař, hranice]
[nálet, zničit, nemocnice, lékař, hranice, sýrie, být, rus, znít, turecko]
Detected events
Detected clusters of event candidates
.@68Jagr má narozeniny: Co dělal ve stejný den před 10, 20 a 30 lety?
(2)
[@68jagr, narozeniny, dělat, stejný, den, 10, 20, 30, rok]
[jágr, narozeniny, dělat, stejný, den, 10, 20, 30, rok, isportzc]
Event candidates

zpravy.rozhlas.cz @zpravyrozhlascz · 4 hod.
Česko začalo jednat o přesídlení první 30 uprchlíků z Itálie a Řecka, do konce roku 2017 by mělo přijmout 2 500. rozhl.as/e0

Aktuálně.cz @Aktualnecz · 2 hod.
Nálet zničil na severu Sýrie nemocnici podporovanou Lékaři bez hranic zpravy.aktualne.cz/zahranici/lete...

Lidovky.cz @lidovky · 2 hod.
Nálet v Sýrii zasáhl nemocnici podporovanou Lékaři bez hranic ldvk.cz/MFy

novinkycz @novinkycz · 1 hod.
Nálet zničil nemocnici Lékařů bez hranic v Sýrii. Byli to Rusové, zní z Turecka novinky.cz/zahranicni/bli...

Denik Sport @DenikSport · 1 hod.
.@68Jagr má narozeniny: Co dělal ve stejný den před 10, 20 a 30 lety? isport.cz/clanek/hokej-n...

Kuba Valenta @KubaValenta11 · 1 hod.
Jágr má narozeniny: Co dělal ve stejný den před 10, 20 a 30 lety? | iSport.cz isport.blesk.cz/clanek/hokej-n... prostřednictvím @DenikSport

Fig. 4. Event detection example (time period $T = 2h$ and acceptance threshold $Th = 0.5$). The rectangle on the right contains six tweets that were saved by our acquisition method. The left "bubbles" show the results of our clustering (two groups containing three and two tweets). The representative tweets are chosen (marked by the bold text on the left side) to be presented to the user [8].

6 Contributions

The main contributions of this paper are summarized below:

- proposing an algorithm for real-time tweet acquisition adapted to the Czech Twitter based on the *UserLists*;
- proposing an algorithm for event detection from Czech Twitter;
- preliminary evaluating of the proposed methods on a small Czech Twitter corpus of the events.

7 Conclusions and Future Work

The main goal of this paper was to propose an approach to harvest messages from Twitter in Czech language with high download speed and a method to detect automatically important events. The proposed harvesting method uses user lists to discover potentially interesting tweets to harvest. We have experimentally shown that this method is efficient because it harvests about 6 times more data than the two other approaches provided by the Twitter API. We further created a small Czech Twitter corpus annotated with the events. We evaluated our event detection method on this corpus. We experimentally shown that the results of the event detection are promising.

Our first perspective consists in the extension of the corpus. On the larger corpus, it will be possible to use longer time periods (two, four, six hours, etc.) to detect the events that are written at different speeds, which cannot be detected on the current version. We also plan to use word embeddings for better representation of words in the clustering method instead of the simple bag of words. This representation is much more precious and it should create better clusters of the tweets. We would like also improve clustering method using more sophisticated semantic similarity functions. The whole system is language independent. Therefore, the last perspective consists in evaluation of the system on other (particularly European) languages.

Acknowledgements

This work has been partly supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications.

References

1. Atefeh, F., Khreich, W.: A survey of techniques for event detection in Twitter. Computational Intelligence 31(1), 132–164 (2015)
2. Earle, P.S., Bowden, D.C., Guy, M.: Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics 54(6) (2012)
3. Hrala, M., Král, P.: Evaluation of the document classification approaches. Advances in Intelligent Systems and Computing 226, 877–885 (2013)
4. Huang, S., Yang, Y., Li, H., Sun, G.: Topic detection from microblog based on text clustering and topic model analysis. In: Services Computing Conference (APSCC), 2014 Asia-Pacific. pp. 88–92. IEEE (2014)
5. Java, A., Song, X., Finin, T., Tseng, B.: Why we Twitter: An analysis of a microblogging community. In: Advances in Web Mining and Web Usage Analysis, pp. 118–138. Springer (2009)
6. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 91–101. ACM (2002)
7. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: The good the bad and the omg! Icwsm 11, 538–541 (2011)
8. Král, P., Rajtmajer, V.: Real-time data harvesting method for Czech twitter. In: in 9th International Conference on Agents and Artificial Intelligence (ICAART 2017). pp. 259–265. SciTePress, Porto, Portugal (24-26 February 2017)
9. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 155–164. ACM (2012)
10. Naaman, M., Boase, J., Lai, C.H.: Is it really about me?: message content in social awareness streams. In: Proceedings of the 2010 ACM conference on Computer supported cooperative work. pp. 189–192. ACM (2010)

11. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREc. vol. 10, pp. 1320–1326 (2010)
12. Powers, D.: Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. Journal of Machine Learning Technologies 2(1), 37–63 (2011)
13. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. pp. 851–860. ACM (2010)
14. Yardi, S., Boyd, D.: Dynamic debates: An analysis of group polarization over time on Twitter. Bulletin of Science, Technology & Society 30(5), 316–327 (2010)
15. Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtskỳ, Z., Hajič, J.: Hamledt: Harmonized multi-language dependency treebank. Language Resources and Evaluation 48(4), 601–637 (2014)