Real-time collection of reliable and representative tweets datasets related to news events

Béatrice Mazoyer[1], Julia Cagé[2], Céline Hudelot[3], and Marie-Luce Viaud[1]

[1] Institut National de l'Audiovisuel,
Bry-sur-Marne, France
bmazoyer@ina.fr, mlviaud@ina.fr
[2] Sciences Po Paris
Paris, France
julia.cage@sciencespo.fr
[3] CentraleSupélec, Mathematics interacting with computer science laboratory,
Gif-sur-Yvette, France
celine.hudelot@centralesupelec.fr

Abstract. This paper is part of a wider work studying the co-influences of Twitter and the production of information by traditional media. A strong prerequisite of this study is to collect, with the limitations of the Twitter API, tweets linked to media events that are representative of the real Twitter activity. This paper describes two proposed approaches to handle this important task. The first one, inspired by information retrieval, puts the focus on query formulation. It consists on bridging the vocabulary gap between traditional news articles and tweets, by iteratively modifying the queries sent to the Twitter API depending on the tweets retrieved by previous queries. The second approach consists in streaming a representative sample of all emitted tweets and dynamically clustering them in events. We also discuss approaches to evaluate the collected datasets under the point of view of their representativity of the real activity on Twitter.

Keywords: Twitter, news, tweets retrieval

1 Introduction

How does information propagate online? Does it propagate differently on news websites and on social media? What is the role of social networks and in particular of Twitter in breaking news?

In an ideal world, to compare news production on social media and on mainstream media, one would need the universe during a given period of time (e.g. the year 2017) and a geographical location (e.g. France, the UK or the US) of documents published on the one hand on social media and on the other hand on mainstream media. Unfortunately, given the limitation of the Twitter API, it is not possible for the researcher to capture the universe of the documents (or

tweets) published on Twitter. Does it mean that the research wont be able to answer the previously formulated questions? No. Because, the researcher can rather use a random sample of the documents, as long as this sample is representative. Why do we need representativity?

Assume that you want to answer the following question: what is the probability for a news story broken on social media to make it to the mainstream media? With the entire set of news stories broken on social media, it will be pretty simple to answer this question; but we dont have this data. Now assume that we get access to a subsample of the documents published on Twitter, but that this sample is not representative. For example, assume that this sample of tweets is such that the tweets characteristics (perhaps because the API provides the researcher with documents tweeted by users with more followers) are such that these documents have a higher probability to make it to the mainstream media. Then using this biased subsample will lead the researcher to overestimate the probability for a news story broken on social media to appear on mainstream media.

The same issue will arise if the researcher wants to tackle the follow-up question: what are the determinants of the success of a news story initially broken on social media? Imagine that the researcher is using a selected sample of tweets that is not representative. Imagine for example that this sample of tweets comes mainly from journalists working for a given media, e.g. *Le Monde*, and that, at the same time, within the set of tweets posted by *Le Monde*s journalists, only the successful ones are part of the sample, then the results of the empirical analysis will be biased in favor of *Le Monde*. In other words, when the researcher will study the impact of the company for which the journalist work (independent variable) on the probability for the news story broken on Twitter to make it to mainstream media (dependent variable), the coefficient obtained for *Le Monde* will ovestimate the real causal impact of the company.

It seems very difficult to correct for this bias. Hence the necessity to have a representative sample of tweets, i.e. a sample of tweets such that the tweets included in our sample do not differ from the tweets that are not included along all the dimensions that may have a direct impact on the dependent variable of interest.

In this paper, we aim at collecting, in real-time, tweets linked to French news that are representative of the total Twitter activity concerning media events. For now, we only work on tweets in French, which is an advantage in terms of volume. Indeed, since the Twitter API puts high restrictions on the volume of tweets returned, we get a largest proportion of emitted tweets by working on a language that is not very represented on Twitter (French tweets represent 1.8% of all emitted tweets). Based on the different ways of collecting tweets with the Twitter API, we explore two types of approaches to collect tweets:

1. *Vocabulary-constrained collection* based on keywords extracted from AFP dispatches. We use the stream of dispatches from the French news agency "Agence France Presse" (AFP). This news agency, like Reuters or Associated Press (AP), has a large network of journalists in many countries, that covers

a wide range of news topics. Previous research has shown that this source covered 95% of all events covered by French news media in 2013 [1].

2. *Random tweets collection* followed by clustering. To get a continuous stream of random tweets, we studied Twitter Sample API[4]. Indeed, previous works show that the tweets from that API are "a representative sample of the true activity on Twitter"[2]. However, the number of French tweets in this sample is very small (2400 tweets per hour on average). It is likely that small events do not generate enough French tweets to be represented in that sample. We thus describe, in Section 3.3, an approach to increase this sample of French tweets.

2 State of the art

Collecting representative sets of tweets means performing together the tweets collection and the evaluation of its representativity. We therefore present in this part: (1) techniques to build sets of relevant tweets and (2) methods to evaluate tweets sets. A tool designed to collect representative tweets sets should comprise both aspects.

2.1 Query building strategies

To our knowledge, few approaches have been proposed in the literature to perform targeted tweet collection based on news articles. Indeed, the literature related to the task of linking tweets and news [3–8] mainly used existing datasets, mostly in English. These articles do not address the issue of real-time collection of representative tweets. Becker et al. [9] do not directly work on news articles, but they define query building strategies based on the title, description, time and location of a selected set of expected events to collect related tweets. Tanev et al. [10] extract 1-grams and 2-grams from the title and first sentence of news articles, and weight them depending on their IDF in a one million news articles collection. They build queries out of these n-grams and explore several query-expansion strategies based on co-occurrences in their news articles corpus. Ning et al. [11] build chains of articles concerning the same event and first collect tweets containing the articles' urls, then extract a list of top ten keywords from those tweets and collect tweets containing these keywords. This method is an attempt to bridge the vocabularies between tweets and news by using keywords from the tweets. Castillo et al. [12], that share common objectives with our work, also use urls to retrieve tweets related to news articles. However, our observations show that most tweets containing a link to an article share only the title or first sentence of that article, without additional vocabulary. This approach is restrictive and may lead to miss a large part of Twitter activity concerning news. Our work aims at building queries that contain Twitter-specific terms in order to achieve higher recall than previously introduced methods, without losing precision.

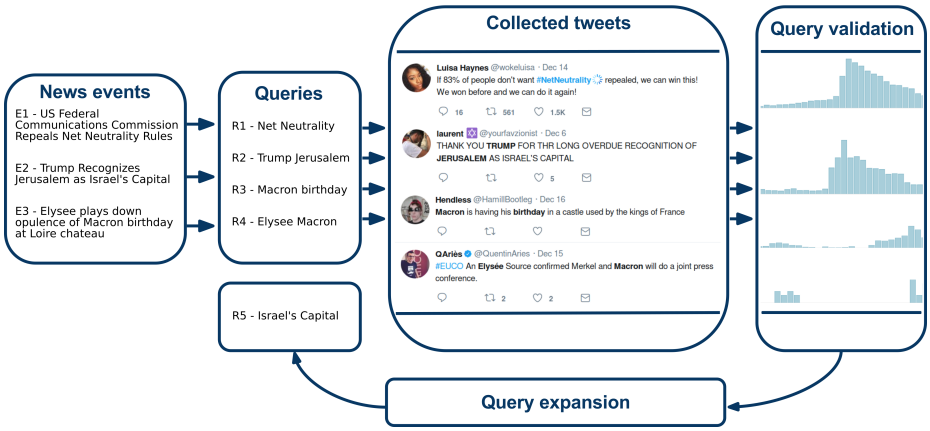
[4] https://developer.twitter.com/en/docs/tweets/sample-realtime/api-reference/get-statuses-sample

2.2 Representativity of tweets collection

Few papers address the problem of the representativity of the collected tweets: we could only find three studies addressing this issue [13, 2, 14]. Morstatter et al. and Wang et al. [13, 14] propose methods to evaluate the Twitter Filter API[5]. This API returns a stream of tweets that match one or several predicates given as parameters. Both articles point out some biases in the collected datasets. In another article [2], Morstatter et al. evaluate the Sample API and conclude to its representativity of the global Twitter activity. However this study does not consider methods to collect larger representative datasets in a specific language as we do for French in this paper.

3 Methodology

Fig. 1. Diagram of our keywords mining approach on three recent events



We present here two approaches for tweet collection:

- A tool designed to build queries in real time based on the stream of AFP dispatches and expand them using Twitter specific vocabulary. This approach is novel with respect to existing works [9–11] since the step of query expansion using word embeddings, while developed in other contexts such as web retrieval [15, 16], has, as far as we know, never been used in the context of tweets retrieval.
- A method to get a significant and representative set of random tweets in French. Our work here aims at maximizing the volume of our random sample, while keeping its distribution characteristics the closest to the one of the

[5] https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter

random sample collected with Twitter Sample API. Events will then be detected by clustering following the same procedure as [17].

3.1 Vocabulary-constrained collection

Our query building tool is composed of several modules designed to (1) extract relevant keywords from AFP dispatches and build queries with them, (2) expand those queries using Twitter vocabulary, (3) validate that these queries return tweets linked to the event. All steps are summarized in Figure 1.

AFP dispatches comprise a title, a timestamp and a body containing a few lines of text describing a news event. 900 dispatches are published every day in average. We define a news event very simply by a new AFP dispatch with a title containing less than k words in common with the titles of previous dispatches published in a time window T. We do not proceed to more complex dispatch clustering since our observations show that when AFP journalists publish several dispatches on a given event, they usually keep the same title and only update the main text with additional details. In practice we have set k to 3 and T to 24 hours. For each event, we aim at building a set of queries that will retrieve tweets related to that event.

Keywords extraction For a given dispatch, we extract named entities from the title and the body. We then build queries using combinations of those entities following a number of rules: two locations cannot form a query (to avoid queries like "Alep Syria"), a query must contain at least two entities and less than four entities. This method is quite similar from what is done by Becker et al. [9]. We also take the exact title of the dispatch as a query, if it contains no named entities. The tweets containing these queries are then collected.

Query expansion We combine several methods to build new queries from collected tweets:

- Urls extraction: like Ning et al. [11], we extract urls from collected tweets and extend our query set with these urls.
- Keywords detection: we build a TF-IDF matrix from all terms of the tweets collected in the past 7 days. All tweets related to the same event form a document. We pick terms with a TF-IDF score higher than a defined threshold (different for 1-grams, 2-grams and 3-grams)
- Synonyms: We use the CBOW model [18] with negative sampling [19] to build vector representations of words in Twitter vocabulary and find potential "synonyms" (terms used in very similar contexts) to expand our first queries. We train the model on a sample of French tweets collected with the Sample API in the past 7 days to get a vector representation of Twitter words. We define as "synonyms" of a word the words that have a cosine similarity higher than a certain threshold with that word-vector. For instance, using 0.85 as a threshold, we get "fh" (abbreviation for "François Hollande") as synonym for "hollande". We then build new queries by replacing terms of previous queries with their synonyms if they have some.

Query validation The previous steps allow us to combine a large number of words to build queries that might be related to the given event. However, these methods can also produce wrong queries, either not precise enough ("Elysee Macron" for example, since l'"Elysée" is the common name for the French president's office), or clearly pointing to something or someone else than the entities extracted from the original dispatch. For instance, the synonyms of "sarkozy" are "sarko" (which is correct), "juppe" and "fillon" (who are politicians from the same party as Nicolas Sarkozy, but not the same person). We therefore need a method to filter incorrect queries.

We can make the following hypothesis: if a news event is discussed on Twitter, the tweets containing words related to that news should form a peak in a time window of a few hours before or after the time of the event. If the tweets collected through a certain query do not peak around the event's time, it is either because the news is not discussed on Twitter, or because the query is not linked to the event in question. Starting from that assumption we are working on an algorithm to detect such anomalies on the time series of tweets and thus remove irrelevant queries.

Iteration The keywords extraction and query validation steps are repeated to increase the number of tweets linked to the considered event, until the set of collected tweet does not provide any new query.

3.2 Evaluation of the tweets collection tool

To the best of our knowledge, there is no dataset corresponding to our task. We are currently designing a user interface to manually label tweets. This interface will allow two types of evaluations:

- Precision: given the text of an AFP dispatch and the list of tweets collected by our tool as presumably linked to the dispatch, the user will assess if each tweet is correctly attributed.
- Recall: the user has to manually formulate queries to retrieve as many tweets as possible in relationship to a given AFP dispatch, and assess if the returned tweets are indeed linked to the dispatch. This method, however, is only a partial solution to the problem of evaluating the proportion of tweets not found by our tool, since the user will not necessarily find all possible terms to query relevant tweets. Another way of approximating the recall would be to compare the proportion of tweets in each collected set to the size of clusters identified with our second approach.

3.3 Random tweets collection

Twitter Sample API returns 1% of the world stream randomly selected[2], among which 1.8% on average are in French. 1% of the French stream represents in average only 2400 tweets an hour. A previous article [17] has estimated the

proportion of news-related documents in tweets corpora to less than 0.2%. If this ratio is the same for French tweets, it means that we would receive between 4 and 5 news-related tweets in French per hour by using the Sample API, a volume much too small for our goals.

Using the Filter API on "neutral" terms with "French" as language parameter allows us to collect a larger volume of tweets in French, but Twitter does not communicate about the tweets distribution. Since we need to ensure that the stream of collected tweets is representative of the tweets emitted on Twitter at the same moment, we designed some methods to compare it with the stream from the Sample API.

Joseph et al. [20] compare several samples collected with the Filter API on the same keywords using different connection tokens[6]: they find that two samples taken at the same time with the same keywords as inputs are "nearly identical" (96% of the tweets are the same, and the sets of tweets that are not have a very similar structure in terms of number of hashtags, urls and mentions per tweet). It is thus not possible to use several connections with the same keywords to get a higher number of tweets. However, spreading different keywords over several API connections should return a higher number of tweets. If these tweets are representative of the real activity on Twitter, the distribution of words within it should be the same (proportionally) to the one of tweets from the Sample API. To ensure that it is the case, we worked on optimizing the collect parameters (number of connections, number of keywords, distribution of keywords over the different tokens) by performing the steps described below:

Finding the most frequent keywords on Twitter. We collected tweets from the Sample API during three months and selected French tweets (retweets excluded). After removing punctuation, capital letters and non-Latin characters, we built a list of the 200 most frequent words in the French Twitter vocabulary.

Clustering keywords depending on their co-occurrences. We built a matrix of co-occurrences of the first 50, 100 and 200 most frequent words in the sampled tweets, and used it to cluster keywords in n clusters ($n \in [2, 4]$). By doing so, we aimed at putting together terms that are frequently used together, to collect sets of tweets with the smallest possible intersection. In total, we had 9 ways of collecting tweets (50, 100 or 200 words, spread over 2, 3 or 4 clusters). To control that our clustering approach was the best, we also tried to randomly spread the same number of words (50, 100 or 200) over the same number of Twitter access tokens (2, 3 or 4). We ran each test during 24 hours and compared the returned tweets with those obtained using the Sample API during the same period of 24 hours.

[6] To use the Twitter API, a connection token is required. Twitter limits the access to its data by generating only one connection token per Twitter account. The total number of tweets that one can get with only one token is limited to 1% of the global tweets volume at a given moment.

Running tests simultaneously. In order to select the best collection method we had to run series of tests in parallel: it is not possible compare tests conducted on different periods of times since the differences between the results could be due to a different tweets distribution between periods. However, we only had 13 Twitter access tokens, and thus could not run all tests simultaneously. Since a collection method requires 2 to 4 tokens, we could only run 6 to 3 tests in parallel.

3.4 Evaluation of random tweets collection

We used Kullback-Leibler divergence to compare the words distributions of the set of collected tweets to the distribution from the Sample API. The best collection method should have a words distribution very similar to the distribution of words collected with the Sample API, and thus the Kullback-Leibler divergence between the two distributions should be close to zero. We also evaluated our collection method with regards to the volume of collected tweets: we used a simple ratio between the volume collected with each method and the volume collected using the Sample API during the same period.

4 Results

Fig. 2. Daily evolution of the KL divergence between each collection method and the Sample API

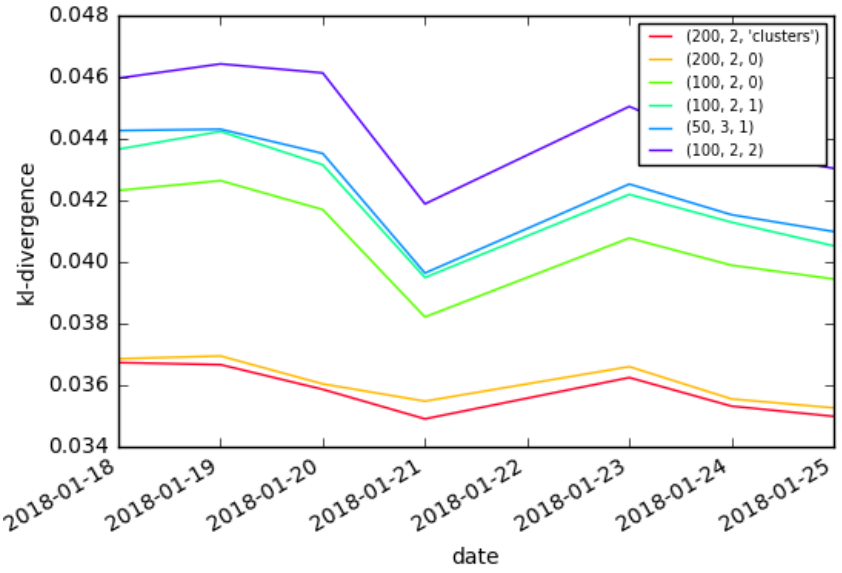
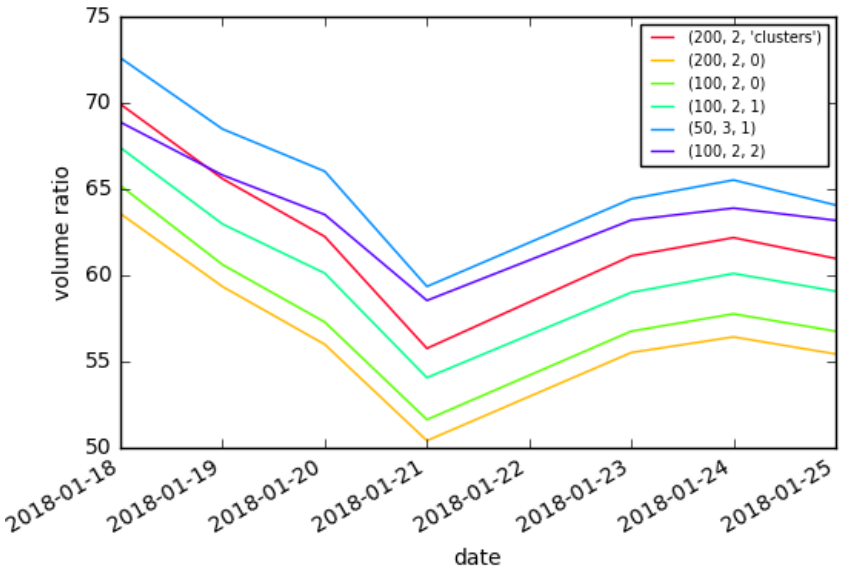


Fig. 3. Daily evolution of the ratio between the volume of collected tweets with each collection method and the volume collected with the Sample API



Note: we have no results on the 01/22, since the connection to the API was broken.

4.1 Tweets collection tool

As explained in Section 3.2, we lack a labeled test-set to perform a correct evaluation of our tweets retrieval method. Our work so far has been directed towards the implementation of a robust architecture to stream and analyze a high number of documents in near real-time. The development of our test interface will allow us to provide concrete results in the near future. We also plan to compare our strategy with the state of the art, presented in section 2.1.

4.2 Random tweets collection

Our results are displayed in Figures 2 and 3. We get consistent results over time: the collection method using words clustered by co-occurrences performs better in terms of proximity to the Sample than other methods using random distribution of words. The volume of collected tweets greatly differs from one collection method to another (from 50 to 75 times higher than the Sample), but the trend also remains stable over time[7]. However, these results need to be

[7] The drop in KL-divergence and in volume of collected tweets on the 01/21 could be linked to the fact that it was a Sunday, but we need to reproduce the experiment over a longer period to confirm that this drop is not a coincidence.

reproduced over a longer period to be confirmed, and we need to conduct wider tests on all collection methods.

5 Related work

5.1 Detecting events in the Twitter stream

A large number of methods have been proposed for detecting "stories" [21] or "events" [22–26] in a continuous stream of tweets. However these methods do not link the detected clusters to news articles. Sankaranarayanan et al. [27] use handpicked "seeders" (Twitter accounts that are known to publish news) and train a naive Bayes classifier to filter news tweets from "junk" tweets. News tweets are then clustered based on TF-IDF. Liu et al. [17] also adopt a two-steps approach to detect news clusters in a continuous stream of tweets. First, they use a noise-filtering algorithm based both on a set of filtering rules and on information credibility features [28]. Then, they perform a clustering step depending on a similarity metric containing named entities, verbs and common nouns. These methods do not treat the problem of linking the collected tweets to news articles, but they are interesting to perform a first selection of relevant tweets.

5.2 Topic Models

Many methods designed to link tweets and news are based on topic models. A topic model takes a collection of documents as input and returns a set of topics and their distribution within the collection (a document is considered as a mix of several topics). Latent Dirichlet Allocation (LDA) [29] is currently the most commonly used topic model. Zhao et al. [4] perform LDA on a collection of news articles and an LDA-like algorithm on a collection of tweets and match the two sets of topics based on the similarity of words distribution. Other works develop algorithms to jointly learn the topics of tweets and news datasets [7], [8]. Guo et al. [5] design a more specific latent variable model including features like hashtags, named entities and temporal relations to link each tweet of their dataset to the closest news article. The previous works perform well on aligning tweets to related news articles, but they are based on static datasets and do not tackle the dynamic nature of news events.

In 2006 Blei and Lafferty proposed Dynamic Topic Model (DTM) [30] a model able to discover the evolution of topics over time. Mele et al. adapted this model to news events [6] and used clustering to link the documents covering the same event across several news streams (news websites, RSS feed, Twitter accounts of media outlets).

Overall, several works address the task of linking tweets and news but they rely either on static datasets or on stream of tweets from manually selected users. We could not find any approach questioning the representativity of the collected tweets. Moreover, most approaches treat tweets and news articles only as text documents and do not take their multimodal nature (urls, mentions of users, pictures, videos) into account.

6 Conclusion and future work

In this paper, we present two types of methods to collect tweets related to news events. The first one is a query formulation approach, that consists in generating potential queries related to an event and using the time repartition of collected tweets to detect and reject wrong queries. The second one is an approach based on clustering randomly collected events. We detailed the tests used to ensure that the collected tweets have a close to random distribution and showed from our first results that neutral words clustered by co-occurrence tend to perform better.

In future works, we plan to develop a user interface that will enable the labeling of numerous tweets, and give us a way of assessing the performance of our search tool. We will also work on methods to dynamically link tweets to news events, with a focus on processing tweets not only as text documents, but rather as multimodal objects.

References

1. Cagé, J., Hervé, N., Viaud, M.L.: The production of information in an online world: Is copy right? CEPR Discussion Paper #12066 (2017)
2. Morstatter, F., Pfeffer, J., Liu, H.: When is it biased?: assessing the representativeness of twitter's streaming API. In: WWW, Companion Volume. (2014) 555–556
3. Kothari, A., Magdy, W., Darwish, K., Mourad, A., Taei, A.: Detecting Comments on News Articles in Microblogs. In: ICWSM. (2013)
4. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing Twitter and Traditional Media Using Topic Models. In: ECIR. (2011) 338–349
5. Guo, W., Hao, L., Heng, J., Diab, M.: Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media. In: ACL. (2013) 239–249
6. Mele, I., Bahrainian, S.A., Crestani, F.: Linking News across Multiple Streams for Timeliness Analysis. In: CIKM. (2017) 767–776
7. Hu, Y., John, A., Wang, F., Kambhampati, S.: ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback. (2012)
8. Hua, T., Ning, Y., Chen, F., Lu, C.T., Ramakrishnan, N.: Topical Analysis of Interactions Between News and Social Media. In: AAAI. (2016) 2964–2971
9. Becker, H., Chen, F., Iter, D., Naaman, M., Gravano, L.: Automatic Identification and Presentation of Twitter Content for Planned Events. In: ICWSM. (2011)
10. Tanev, H., Ehrmann, M., Piskorski, J., Zavarella, V.: Enhancing Event Descriptions through Twitter Mining. In: ICWSM. (2012)
11. Ning, Y., Muthiah, S., Tandon, R., Ramakrishnan, N.: Uncovering News-Twitter Reciprocity via Interaction Patterns. In: ASONAM. (2015) 1–8
12. Castillo, C., El-Haddad, M., Pfeffer, J., Stempeck, M.: Characterizing the life cycle of online news stories using social media reactions. In: CSCW. (2014) 211–223
13. Morstatter, F., Pfeffer, J., Liu, H., Carley, K.M.: Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In: ICWSM. (2013)
14. Wang, Y., Callan, J., Zheng, B.: Should we use the sample? analyzing datasets sampled from twitter's stream api. ACM Trans. Web **9**(3) (June 2015) 13:1–13:23

15. Diaz, F., Mitra, B., Craswell, N.: Query Expansion with Locally-Trained Word Embeddings. CoRR abs/1605.07891 (May 2016)
16. Roy, D., Paul, D., Mitra, M., Garain, U.: Using Word Embeddings for Automatic Query Expansion. CoRR abs/1606.07608 (June 2016)
17. Liu, X., Li, Q., Nourbakhsh, A., Fang, R., Thomas, M., Andersony, K., Kociubay, R., Veddery, M., Pomervilley, S., Wudali, R., Martiny, R., Dupreyy, J., Vachhery, A., Keenan, W., Shah, S.: Reuters Tracer: A Large Scale System of Detecting & Verifying Real-Time News Events from Twitter. In: CIKM. (2016) 207–216
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781 (2013)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. (2013) 3111–3119
20. Joseph, K., Landwehr, P.M., Carley, K.M.: Two 1% s Don't Make a Whole: Comparing Simultaneous Samples from Twitter's Streaming API,. In: SBP. (2014) 75–83
21. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: HLT-NAACL. (2010) 181–189
22. Atefeh, F., Khreich, W.: A Survey of Techniques for Event Detection in Twitter. Computational Intelligence **31**(1) (February 2015) 132–164
23. Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: SIAM. (2012) 624–635
24. Weng, J., Lee, B.S.: Event Detection in Twitter. In: ICWSM. (2011)
25. Liang, S., Yilmaz, E., Kanoulas, E.: Dynamic Clustering of Streaming Short Documents. In: KDD. (2016) 995–1004
26. Zhang, C., Liu, L., Lei, D., Yuan, Q., Zhuang, H., Hanratty, T., Han, J.: TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams. In: KDD. (2017) 595–604
27. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: TwitterStand: news in tweets. In: GIS. (2009) 42–51
28. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: WWW. (2011) 675–684
29. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan) (2003) 993–1022
30. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: ICML. (2006) 113–120