# Efficient Document Filtering Using Vector Space Topic Expansion and Pattern-Mining

## The Case of Event Detection in Microposts

Julia Proskurnia
École Polytechnique Fédérale de Lausanne, Switzerland
julia@proskurnia.in.ua

Ruslan Mavlyutov
University of Fribourg
Fribourg, Switzerland
ruslan@exascale.info

Carlos Castillo
Universitat Pompeu Fabra
Barcelona, Spain
chato@acm.org

Karl Aberer
École Polytechnique Fédérale de Lausanne, Switzerland
karl.aberer@epfl.ch

Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

## ABSTRACT

Automatically extracting information from social media is challenging given that social content is often noisy, ambiguous, and inconsistent. However, as many stories break on social channels first before being picked up by mainstream media, developing methods to better handle social content is of utmost importance. In this paper, we propose a robust and effective approach to automatically identify microposts related to a specific topic defined by a small sample of reference documents. Our framework extracts clusters of semantically similar microposts that overlap with the reference documents, by extracting combinations of key features that define those clusters through frequent pattern mining. This allows us to construct compact and interpretable representations of the topic, dramatically decreasing the computational burden compared to classical clustering and k-NN-based machine learning techniques and producing highly-competitive results even with small training sets (less than 1'000 training objects). Our method is efficient and scales gracefully with large sets of incoming microposts. We experimentally validate our approach on a large corpus of over 60M microposts, showing that it significantly outperforms state-of-the-art techniques.

## CCS CONCEPTS

•**Information systems → Content analysis and feature selection; Document filtering;** *Social networks; Similarity measures; Clustering and classification;*

## KEYWORDS

Event detection, microposts, frequent patterns mining, semantic attributes.

## 1 INTRODUCTION

Social media—and in particular Twitter—have reshaped the news industry. Billions of users (2.8 Bn - wearesocial.com) contribute live updates on events that are happening in their vicinity using various social media platforms, thus allowing not only journalists but also citizens or stakeholders to follow breaking news in near real-time. A number of activities such as journalism, activism, or disaster recovery can be facilitated by means of social media [39]. However, *extracting relevant information from social media in a reliable and efficient manner* still remains a challenge.

In this paper, we tackle the problem of efficiently identifying documents that are relevant to a given query. A query in our context is represented by a small set of textual documents that are relevant to a specific topic of interest. As a particular instance of this problem, we focus on *extracting microposts that are relevant to a given event* in the following.

Several methods have been proposed for this problem, we summarize them in Section 2. One approach is to apply some semantic matching (e.g., edit distance or lexical overlap) between the description of the event and a series of microposts. In general, such methods identify messages that are similar to the query of interest, but are computationally expensive and yield a poor recall, i.e., fail to produce comprehensive results. Another approach is to leverage knowledge bases, thus taking into account semi-structured or unstructured descriptions of well-known entities and events in the matching process. In such approaches, however, domain-specific knowledge is generally underrepresented. Finally, a number of classification and clustering approaches have been proposed recently for this problem. These approaches are typically computationally expensive, require a well-defined and accurate metric of similarity between two texts, and usually require a large corpus of labeled data, thus limiting their potential domain of application.

We propose a novel methodology that is both efficient and requires a very small labeled training set while performing on par with methods that utilize much larger training datasets or that are computationally very expensive. Specifically, we propose a technique based on frequent itemsets (patterns) extracted from the query. We measure the distance between various query items to extract the patterns. Specifically, we leverage text similarity metrics that rely on word embeddings that are pre-trained on very large

*CIKM'17, November 6–10, 2017, Singapore.*

ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133016

collections of microposts. Our solution is task-independent and is evaluated on the complex task of event extraction from social media streams. We show that our method outperforms state-of-the-art baselines (lexicon, embedding similarity, k-nearest neighbors and classification based on word embeddings) and that is it computationally efficient compared to instance-based (k-NN) approaches. Broadly speaking, we show how syntactic or semantic clustering can be efficiently replaced by semantic pattern extraction for event detection.

**Our contribution.** We present a new method for filtering microposts that match a specific query. The query is a textual description of the topic of interest; in our running example and in our experiments, this topic of interest is an event. Based on this description, our method automatically generates a small *seed set* of microposts, based on text similarity. Then, we apply frequent pattern (itemset) mining on the seed set. Among the extracted patterns, we select those that are associated with semantically homogeneous groups of microposts. These patterns (called topical patterns) are then compared to an incoming stream of messages in order to select all microposts matching the query. Our technique is presented in detail in Section 4.

In summary:

- we describe an efficient solution requiring minimal annotations to filter and identify microposts that are relevant to a given query;
- we present an extensive evaluation with multiple baselines and show that our approach outperforms them over a large dataset of 3TB of Twitter messages spanning two years;
- finally, we release the source code of our technique as well as a collection of annotated event messages for different classes of events.

The rest of the paper is organized as follows. We start with an overview of related work in Section 2. We describe our process for collecting the data from social media as well as identifying seed microposts related to the events in Section 3. We present our topical document extraction model and compare it to existing models in Section 4. We experimentally evaluate the models and discuss them in Section 5. Finally, we summarize our results and outline future work in Section 6.

## 2 RELATED WORK

Extracting online content based on a query is challenging and typically requires large amounts of annotated data to build supervised models [3, 5, 9, 15, 32, 41]. In some cases, the query is not known a priori and is only implicitly represented through a set of documents that are relevant to a topic of interest [17, 18, 20]. Similarity-based approaches tend to be inefficient [8] and difficult to scale.

Another approach to tackle the topical document detection problem is to rely on content clustering and topic modeling (see Table 1). However, these approaches work best for document extraction relating to past events (thus, specific details about the event are known and can be used for the extraction) and are hard to adapt to a stream processing context (where neither particular details nor dates are known ahead of time). A number of techniques leverage a lexicon that can effectively and accurately represent a given topic, yielding a high precision but a rather low recall. Olteanu et al. [27], for instance, uses pseudo-relevance feedback to improve recall for the lexicon-based methods, which however hampers their capacity to detect new events [31]. [38] leverages semantic analyses and ontologies to detect complex events with a high precision. Finally, a range of new deep learning architectures have been recently proposed to both represent the document in a semantic space as well classify the documents by topics based on their vector space representation [17, 18, 20]. Such methods are supervised and require a large corpus of annotated data.

Contrary to the various methods described in Table 1—such as classification methods requiring a substantial amount of annotated data, or methods based on query similarity that require pairwise similarity comparisons between the query text and the input data—we propose a method that is more efficient and accurate, and achieves high performance even with very small training sets.

## 3 DATA COLLECTION AND SEED EXTRACTION

In this section, we introduce the data sources we use and our data collection process (Section 3.1), explain how seed messages describing the events are extracted (Section 3.2), and describe the data annotation process that is used to evaluate the quality of our results (Section 3.2).

### 3.1 Data Collection

The topics we use in our examples correspond to large-scale events that are covered widely by international media. Specifically, we focus on terrorist attacks: uses of violence to create fear, for ideological purposes, and aimed at civilians or noncombatant targets [24]. We create a database of attacks by integrating information from Wikipedia and from the Global Terrorism Database (GTD).

**Wikipedia data.**[1] We crawled all attacks in 2014 and 2015, which are available on 24 separate pages indexed by month, and contain information on 650 events. This list applies the definition of violence from a non-state actor, without considering the restriction of being against civilians. Hence, three authors of this paper manually annotated the events to discard those perpetrated against combatants or armies. The attack was added to the database when the agreement between the annotators was 100%. A total of 592 events were selected and are listed on Table 2, along with information on the country and type of the attack as described on Wikipedia[2].

**Global Terrorism Database (GTD).**[3] GTD contains over 15K records from the same period, including minor and major incidents involving civilians. The GTD dataset was created to enhance the initial descriptions we obtained from Wikipedia. We use GTD and Wikipedia attack descriptions as the input queries.

**Twitter data.** We performed a rate-limited data collection from Twitter, collecting up to 5%[4] of all microposts (tweets) posted during 2014 and 2015 using Twitter's Streaming API. The dataset resulted in over 3TB of data, out of which 60M tweets were posted in English.

[1] https://en.wikipedia.org/wiki/List_of_terrorist_incidents

[2] Information from Wikipedia contains a variety of metadata, including the location and date, a summary of the event, the number of casualties, and the suspected perpetrator.

[3] https://www.start.umd.edu/gtd/

[4] 5 Twitter accounts were requesting Twitter Streaming API during the 2 years.

| *Topical Document Detection Approach* | *Short Description* |
|---|---|
| ***Retrospective Topic Detection*** | |
| Feature engineering | [5, 32, 35, 41] represent both the input stream messages or their clusters through a variety of features, e.g., term frequencies and weights, topicality, skewness, timeliness, periodicity, keyword position, context etc. [26] further stratifies the topics into sub-events based on four main features, contents, time, diffusion degree and sensitivity. [2] shows that NLP-based lexicons work best for specific topics. [21] estimates the importance of classified tweets for a particular topic represented as an event. [34] iteratively selects phrases to track a particular topic and thus improves the extraction over time. |
| Content clustering | [28] surveys various clustering techniques to identify topical events on the web. [16, 44] leverage co-occurring words to identify topical events. [4, 40] first cluster semantically close tweets and then extract event-specific features from the clusters. [25] describes a production-ready system based on text similarity clustering and cluster burst detection for event detection. [12] clusters keywords based on their spectral representation using Kullback-Leibler divergence. [11] uses LSH to make document clustering more efficient and then inspects each bucket separately to identify the topic of the event. [10] uses LDA by leveraging the proximity of the tweets as well as the source of the message to cluster the tweets. |
| Similarity-based ranking | [18] compares various similarity metrics based on document vector representations and shows that averaging the embeddings in a document leads to underestimating the similarity between documents. A better measure of similarity is defined as Word Mover's Distance that is explored further in [17]. [43] explores various embedding estimations of the queries for a specific topic extraction. [14] utilizes Web-click graphs to rank documents for a given query. |
| ***Unseen Topic Detection*** | |
| Clustering, TFIDF, LDA | [23] leverages TFIDF for document similarity to further filter and enhance event detection. Along similar lines, [17] shows that TFIDF similarity metrics perform on par with expensive Word Model Distances thus allowing scalable and easy to implement alternatives for initial document extraction. [31] proposes an accurate open domain event extraction pipeline that gathers named entity, event phrase (CRF), date, and type (LinkLDA). [38] uses semantic analyses and ontologies to detect complex events with a high precision. [30] proposes an efficient LSH-based heuristic to detect new events. [13] explores user profiles and interests to trace specific topics. [19] uses auxiliary word embeddings to model topic distributions in short texts. [33] relies on non-parametric distributional clustering to infer topical infection of the users in information cascades. [29] uses LDA to infer a central topic model that is further enhanced with a two-phrase random walk, thus allowing to accurately model even-specific topics. |
| Classification | [3] handles event detection as a multi-task learning problem and proposes an optimization that utilizes the tweet contents and categorical relations. [9] relies on non-parametric topic modeling within time epochs to track semantically consistent topics and models event arrivals as a Poisson process with (non) bursty periods. [15] explores linear models with a rank constraint and a fast loss approximation and shows that they perform on par with deep learning classifiers. |
| Dataless Text Classification | [20] learns to extract relevant documents based on a small seed of related keywords by exploiting explicit word co-occurrence patterns between the seed words and regular words. Similar extraction techniques leveraging lexicon expansion are described in [27]. [7, 22] analyze the extent to which query words can be used to represent a topic of interest for further extraction. [37] shows how semantic representations of a query and a document allow to accurately measure the similarity between the two. |

**Table 1: Overview of the most prominent approaches and applications of topic tracking on social media.**

| Country | Bombing | Attack | Shooting | Raid | Events | Tweets |
|---|---|---|---|---|---|---|
| Iraq | 84 | 2 | 0 | 0 | 90 | 811 |
| Israel | 3 | 55 | 9 | 13 | 84 | 1001 |
| Nigeria | 48 | 5 | 6 | 0 | 72 | 2408 |
| Afghanistan | 42 | 3 | 3 | 0 | 53 | 657 |
| Pakistan | 39 | 7 | 4 | 0 | 51 | 3189 |
| Egypt | 25 | 2 | 3 | 0 | 31 | 344 |
| Yemen | 19 | 2 | 0 | 0 | 22 | 175 |
| Syria | 21 | 0 | 0 | 0 | 21 | 483 |
| Somalia | 16 | 1 | 1 | 0 | 18 | 251 |
| Cameroon | 10 | 0 | 4 | 0 | 15 | 76 |

**Table 2: Wikipedia dataset characteristics. We list the top 10 countries and the top 4 attack types as described on Wikipedia. The column "Event" corresponds to the total number of events for a country, while "Tweets" contains the number of matching microposts for each attack description.**

We relied on the NLTK python library[5] to detect the language. This is the dataset over which we all extraction techniques are evaluated in the following.

### 3.2 Seed extraction

Our method leverages a small set of seed microposts (training dataset) that are later used to extract patterns to determine which microposts should be selected. This training dataset is directly provided to the method described in Section 4. Since we have at our disposal a database of attacks along with their description (see above), we use this information to the seed microposts. As shown in [18], TF.IDF-based similarity metrics perform on-par with complex semantic similarities. Thus, to extract an initial set of relevant microposts, we apply a TF.IDF-based algorithm. The algorithm identifies whether a micropost describe any of the attacks in the database.

To set the similarity threshold $\theta$ between the event descriptions and the microposts, we manually annotated (as described in the Annotation paragraph below) a random sample of 300 tweets related to some attack for various thresholds. As a result, we picked the threshold to $\theta = 0.27$ as this value yields the best precision (95%) on our sample.

In total, we obtained 17'093 seed microposts related to terrorist attacks. Table 3 shows some examples of attack descriptions and the related microposts.

**Data annotation** We adopt a consistent process to annotate the microposts and to determine the quality of our results (Section 5). Specifically, two authors of this paper manually annotated the relevance of the microposts selected by the algorithm (or by any of the baselines).

## 4 METHOD DESCRIPTION

This section describes the method we propose to filter relevant microposts for a given query. Our method first represents each

[5] http://www.nltk.org/

| Terrorist attack description | TF.IDF similarity | Extracted tweet |
|---|---|---|
| A suicide bomber attacked a police academy in 5th police district, Kabul city, Kabul province, Afghanistan. In addition to the suicide bomber, 25 people were killed and 25 others were wounded in the blast. The Taliban claimed responsibility for the incident. | 0.30 | #KCA #VoteJKT48ID guardian: Taliban attack parliament building in Kabul with suicide car bomber and RPGs |
| Assailants opened fire on Dr. Waheedur Rehman in Dastagir area, Karachi city, Sindh province, Pakistan. Rehman, a Karachi University professor, was killed in the attack. No group claimed responsibility for the incident. | 0.33 | F.B Area Block-16 Me Firing Se Karachi University Shoba Ablagh-e-Aama Ke Assistant Professor Syed Waheed Ur Rehman S/O Syed Imam Janbahaq. |
| Assailants abducted seven Coptic Christian Egyptians from their residence near Benghazi city in Benghazi district, Libya. The seven Egyptians were killed the same day. No group claimed responsibility for the incident. | 0.43 | #IS in #Libya claims responsibility for abducting 21 Egyptian #Christians,http://t.co/32l8YCLL35 #Egypt #ISIS |
| Two suicide bombers opened fire and then detonated inside a classroom at the Federal College of Education in Kano city, Kano state, Nigeria. In addition to the two bombers, at least 15 people were killed and 34 others were injured in the blasts. Boko Haram claimed responsibility for the attack. | 0.44 | Boko Haram claims responsibility for Kano bomb blast, share photo of the male suicide bomber: B... |
| A rocket landed inside a community and detonated in Sdot Negev regional council, Southern district, Israel. There were no reported casualties in the blast. No group claimed responsibility for the attack. | 0.56 | #BREAKING: A rocket from #Gaza hit Sdot Negev Regional Council in southern #Israel. No damage, no injuries |

**Table 3: Examples of seed microposts related to the attacks.**

input query by a *seed set*; it mines "topically homogeneous" patterns from the seed set. The patterns are then placed into an index which is used for efficient filtering, i.e., to select the microposts that contain a pattern and are hence relevant to the input query. We give an overview of the whole method in Section 4.1. Next, we describe the text similarity metric (Section 4.2) and pattern extraction approach (Section 4.3). Finally, we compare this approach to alternative clustering methods in Section 4.4.

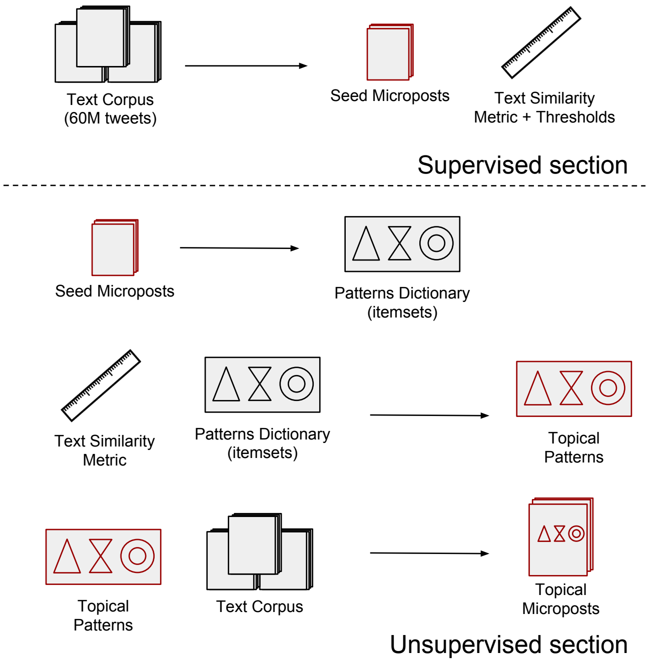

**Figure 1: Pipeline overview**

## 4.1 Overview

Figure 1 outlines the major steps of our approach, which combines two key insights: (1) an appropriate distance metric can be leveraged to estimate their topical similarity between microposts, and (2) we can take the best of two worlds by using pattern extraction techniques to combine both supervised and unsupervised learning.

**Similarity metric adjustment.** The only part of the process that requires human supervision is the selection and the adjustment of the distance metric between documents, which has to be performed once per corpus – in the present case just once as there is a single input corpus containing all microposts.

The metric adjustment assumes the following:

- all possible pairs of documents (microposts) existing in the input corpus belong to one of the following classes: identical $(x, y_{\text{ident}})$, similar $(x, y_{\text{similar}})$, topically related $(x, y_{\text{related}})$, or unrelated $(x, y_{\text{un related}})$;
- there exists a distance metric $d$ that defines the following order on the pairs of documents:

$$d(x, y_{\text{ident}}) < d(x, y_{\text{similar}}),\ d(x, y_{\text{similar}}) < d(x, y_{\text{related}}),$$
$$d(x, y_{\text{related}}) < d(x, y_{\text{unrelated}})$$

If those two assumptions hold, we can determine a threshold $d_{\text{related}}$ that separates pairs of topically related documents from unrelated pairs of documents. Fortunately, there is a large body of literature on this topic and we do not need to invent a new text similarity metric. The threshold value $d_{\text{related}}$ can then be estimated empirically on a validation set, for a target type of documents (in this paper, microposts). Details on this step are provided next in Section 4.2.

**Pattern mining.** We extract frequent patterns (itemsets) from the seed microposts and use them to filter the input to produce a larger set of relevant microposts. Given that there might be many such patterns potentially (with many patterns not representing any relevant subset of microposts), we need to filter those patterns. Towards that goal, we note that *a relevant pattern induces a topically homogeneous set of microposts.*

We call a pattern *topically homogeneous* if all the microposts that it matches are topically related to each other. To measure topical homogeneity, we estimate the expected pairwise distance between a pair of microposts selected by a pattern by randomly sampling pairs of microposts containing the pattern. If the expected distance is lower than a threshold value $d_{\text{related}}$ estimated during the similarity metric adjustment step, then the pattern is considered to be topically homogeneous.

Pattern extraction has several benefits compared to other approaches:

(1) Unlike most of supervised learning approaches, the performance of our method (especially the precision) depends less on the size of the seed. The resulting accuracy of text selection is boosted by the effectiveness of the distance metric and the selected thresholds.
(2) Compared to the instance-based machine learning methods (like k-NN), pattern extraction is more flexible and efficient from a computational perspective. In general, for every new document, k-NN would require computing the distance between this document and all the seed documents, which in the most simple case yields a complexity of $O(|\text{Docs}|\cdot|\text{SeedDocs}|\cdot\text{AvgWordsPerDocument})$ (if we are using distance metrics that only weigh word overlap of the documents). With large training sets and an elaborate distance metric (like the one we are using in this paper), k-NN rapidly becomes impractical. Another important drawback of k-NN is the necessity of a proper set of negative samples. In the context of topic extraction, one needs to create a set of neighboring topics, which is often a very complex task. Our method on the other hand does not require negative samples. The computational complexity of pattern extraction in general is NP-hard, though limiting the length of the patterns and the textual features dramatically limits the number of possible patterns that can be extracted from the seed documents. With topically homogeneous patterns, we reduce the number of elements to take into account to a few thousands even for large seeds. Every pattern is a conjunction of a limited number (maximum 5 in our case) of textual features. In that case, checking whether a document contains at least one topical pattern can be done in sublinear time (in terms of the size of the text) with proper indexing techniques.
(3) Extracted patterns are easy to interpret.
(4) The support values of the patterns can be used to rank the documents with respect to their relevance to a topic.

## 4.2 Text similarity metric

Our method requires a metric for measuring text similarity (see above). We picked Word Mover's Distance (WMD) metric since [18] shows that it performs best for short text semantic similarity. WMD attempts to find an optimal transformation between documents $d$ and $d'$. The method is solving the following linear optimization task with constraints:

$$WMD(d,d') = \min_{T\geq 0}\sum_{i}^{n}\sum_{j}^{m} T_{ij}c(i,j) \qquad \text{subject to:}$$

$$\sum_{j}^{m} T_{ij} = 1/n,\ \forall i \in \{1,\dots,n\}\,, \qquad \sum_{i}^{n} T_{ij} = 1/m,\ \forall j \in \{1,\dots,m\}$$

where:

- $n, m$ are the number of words in documents $d$ and $d'$,
- $T_{ij}$ is the weight of word $i$ (WDM works with nBOW representations of documents, so a word weight is equal to $1/|d|$) from document $d$ that is going to be transferred to word $j$ of document $d'$, and
- $c(i,j)$ is the "traveling" cost between words $d_i$ and $d'_j$.

In Kusner et al. [18], the traveling cost was selected to be equal to the Euclidean distance between vector representations (in the `word2vec` embedding space) of words. According to our experiments, however, the Euclidean distance suffers from the so called "curse" of dimensionality for a high-dimensional vector space (over 100 dimensions) as most of the distances end up having similar values. On the other hand, the cosine similarity empirically yields less skewed distance distributions. Hence, we rely on cosine similarity in the following, and the distance metric between words for our method becomes:

$$c(i,j) = 1 - X(d_i)X(d'_j) \tag{1}$$

where $X(m)$ is the vector representation of word $m$. For our method we use the FastText [20] vector model with a dimensionality of 300.

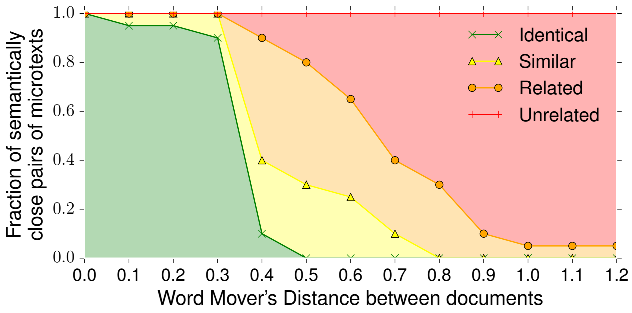

**Figure 2: Relatedness of documents as function of their pairwise distance.**

We now assess how WMD values can be leveraged to identify related documents. To identify reliable threshold values for topical similarity in the WMD space, i.e., to determine $d_{\text{ident}}$, $d_{\text{similar}}$, and $d_{\text{related}}$, we sample document pairs for every WMD value interval from 0.1 to 1.2 with a step of 0.1. The sample sizes were equal to 100, giving us 1'200 document pairs in total. For every WMD interval sample, 2 authors of the paper checked the pairs and labeled them as (1) copies, (2) semantically identical texts, (3) topically related texts, or (4) different texts. The distribution is shown in Figure 2. Based on those results, we selected a WMD value of $d_{\text{related}} = 0.5$ as threshold for topical similarity. A pair of documents with WMD smaller than 0.5 [6] has a probability of more than 90% to be topically related (as it is close to 80% for WMD=0.5 and increases for lower values of WMD).

[6] *Robustness tests of various WDM thresholds against short text filtering as presented in Table 6.* Threshold of 0.4-0.45 results in low recall (about 1.5-2 times less than for 0.5). Threshold of 0.55-0.6 results in 1.5-2 times higher recall (this improvement reduces for larger training sizes) and lower precision. In terms of F1, 0.5 precedes 0.6 for larger number of training examples and vice versa for smaller number of training examples.

## 4.3 Pattern extraction

The problem of mining patterns (or associations) from item sets was introduced in [1]. Pattern extraction from text can be formally introduced as follows: Let $D = i_1, i_2, .., i_m$ be a set of $m$ distinct attributes (we call these attributes *markers* in the context of this paper). Each document in a corpus $T$ has a unique identifier $TID$ and is associated with a set of markers (itemset). As such, it can be represented as a tuple $< TID, i_1, i_2, .., i_k >$. A set of markers with $k$ items is called a $k$-itemset. A subset of length $k$ is called a k-subset. An itemset is said to have a support $s$ if at least $s$ documents in $T$ contain the itemset. An association rule is an expression $A \Rightarrow B$, where itemsets $A, B \subseteq D$. The $confidence$ of the association rule, given as $support(A \cup B)/support(A)$, is the conditional probability that a transaction contains B, given that it contains A.

Originally, the pattern extraction task consists of two steps: (1) mining frequent itemsets, and (2) forming implication rules among the frequent itemsets. In our method we concentrate on the extraction of topically homogeneous itemsets, i.e., patterns that are present in documents that are topically related to each other.

To answer whether a set of documents containing the itemset is topically related, we estimate the mean WMD value between the documents by calculating the average WMD value for a sample of documents pairs. If the resulting value is less than the threshold value for WMD topical relatedness ($d_{\text{related}}$), then we consider the set of documents as being topically related.

**Pattern mining algorithm.** Starting with the full dictionary of markers present in the seed microposts, we use the ECLAT algorithm [42] for pattern mining. ECLAT is a scalable (due to initial parallelization) depth-first search family of pattern mining algorithms. The minimum support of an itemset is defined by a minimum sample size of document pairs that is required to reliably estimate the mean of the pairwise distances between documents that contain the pattern. In our case, we chose this value to be more than 40, so that assuming a normal distribution of pairwise distances we will have enough pairs of documents to estimate the mean distance.

To speed-up the process of pattern extraction, we add two pruning criteria. First, we stop growing topically homogeneous itemsets, since all their supersets will be producing subclusters of the current cluster of topically related documents. We also define a maximum pattern length; in our experiments, we only use patterns composed of at most 5 markers.

**Types of attributes.** For this paper we only use two type of attributes - stemmed and lowercased words presented in the text, and synsets (clusters of semantically similar words) that we describe below. For stemming we use the Porter stemmer. We also remove stop words, since their absence helps to significantly reduce the amount of irrelevant patterns.

**Sets of related words (synsets).** We leverage word embeddings constructed as explained in Section 5.1 to construct sets of related words from our Twitter dataset. We call them "synsets" in the following, but note they are not necessarily synonyms of each other, but closely related words. As shown by Schwartz et al. [36], skip-gram models in combination with cosine similarity yield similarity estimations on par with more complex state-of-the-art techniques.

| Seed | Synset |
|---|---|
| bomb | bombing, bomber, explosives, detonated |
| shot | shooting, shoot, shots |
| kill | kidnap |
| nigeria | kenya, ghana, uganda, benin |
| huge | massive, enormous, tremendous |
| gas | hydrocarbon, combustion, sulfur, methane |

**Table 4: Examples of extracted synsets.**

Two authors of the paper manually evaluated several cosine similarity thresholds ranging from 0.5 to 1.0. A threshold of 0.65 resulted in the most coherent pairwise semantic word proximity. For the 30K most frequent words in the whole Twitter dataset, we construct the synsets greedily in a "snowball" fashion, i.e., for each word we identify a set of most semantically similar words; each of those words is then used in turn to find semantically similar words, and so on. Each word is added to the synset if it is similar to at least 30% of the words that are already there, reducing topic drift. Some examples of synsets are shown in Table 4. On average, synsets have 3.6 terms, with a median of 3 terms.

## 4.4 Patterns vs Clustering: a case of coverage

One may ask whether frequent itemsets cover a significant part of topically-related documents, particularly when compared to potentially higher-recall methods, such as clustering.

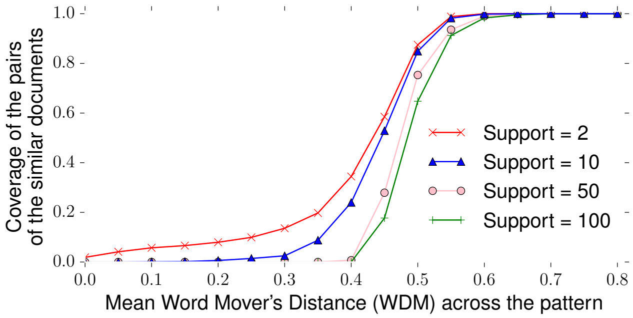

**Figure 3: Estimated part of topically-related document pairs that can be covered by patterns**

To examine this question, we sampled 200K document pairs that were considered at least topically related (WMD value smaller than 0.5). Then, for every pair we looked for itemsets that were included in both documents and had a given support value. The distributions of coverage (percentage of pairs covered by patterns with defined parameters) as a function of minimum pattern support and mean of WMD are shown in Figure 3. We observe that at least 90% of the sampled pairs can be covered with topical patterns with support greater than 10, and approximately 75% of the pairs can be covered with topical patterns with a support greater than 50. This means that in the most pessimistic case, the selected support value of 40 guarantees that we cover at least 75 – 80% of the documents related to the topic.

## 5 EXPERIMENTAL RESULTS

To evaluate our approach, we compare it to several baselines (Section 5.1) in terms of precision and relative recall (Section 5.2); results are summarized in Section 5.3 and discussed in Section 5.4.

### 5.1 Baselines

We compare our methods against a number of state-of-the-art baselines that cover the main approaches for topical document extraction (Section 2). Specifically, we implemented a Pointwise Mutual Information (PMI) based lexicon expansion, and three methods based on word embeddings: a semantic centroid classifier, a FastText similarity ranking technique, and a proximity-based (kNN) method.

These baselines require training data; we use synthetically generated training examples which are nevertheless of high quality. The training examples are obtained using the seed selection method described in Section 3.2, which as discussed is more than 95% precise. We vary the size of the training set available to each baseline $N_{\text{train}}$ from 1 to 10K randomly sampled positive examples. For methods requiring negative examples, we select an equally-sized set of negative examples, which are sampled from all microposts that are not in a seed dataset. The assumption here is that the presence of tweets related to terrorist attacks in the general dataset of tweets is negligibly small, so false negatives will be minimal. However, this heuristic is not appropriate for k-NN, so we had to slightly modify it, as explained below.

For the methods based on word embeddings, we trained a FastText skip-gram model [6] over the 60M English tweets described in Section 3 with default parameters: vector size – 300, window size – 5, negative sampling, minimum words count – 10.

**1. Corpus-based PMI.** In this paper, we used the PMI-based term scoring method described in Olteanu et al. [27] that measures the difference between the relatedness of a term $t$ to (1) an event class $a$ and (2) a non-event class $\neg a$. This is defined as follows:

$$PMI(t) = \log_2 \frac{p(t|a)}{p(t|\neg a)} \qquad p(t|a) = \frac{\text{count}(t, a)}{\text{count}(MARKERS, a)}$$

where $p(t|a)$ and $p(t|\neg a)$ are the probabilities of $t$ appearing in event-related and not event-related microposts, respectively. $MARKERS$ can be any syntactic representation of a text; we use unigrams and bigrams in this evaluation. The top ranked unigrams and bigrams for all the events (terrorist attacks) are shown in Table 5.

**2. Semantic centroid classification.** As a second baseline, we use a linear classifier trained on the semantic representations of the microposts, as described by Kenter and de Rijke [17]. Every word in the training data is represented by a set of 300-dimensional features that correspond to the embedding representation of each word. We derive this feature vector by averaging each dimension of the words in the sentence.

**3. FastText-based similarity ranking.** This approach enhances the previous baseline by learning how to combine word embeddings into a text representation as described by Joulin et al. [15]. Thus, the resulting text representations are better to distinguish topical tweets. To find similar tweets, each short text is sent through the classification model so that task-specific embeddings are obtained. Then, the representations of the target tweets are compared (using cosine similarity) to the unlabelled ones. Several similarity threshold are tested to select the final results; a distance threshold of 1.1 radians results in the best accuracy.

| | | | |
|---|---|---|---|
| Unigrams | bernardino | gunmen | bombers |
| | haram | kano | militants |
| | jerusalem | garissa | mandera |
| | parenthood | baghdad | siege |
| | copenhagen | synagogue | bombings |
| | leytonstone | blasts | tunisia |
| Uni- and bigrams | bernardino | planned_parenthood | kano |
| | san_bernardino | copenhagen | garissa |
| | haram | in_peshawar | baghdad |
| | boko_haram | leytonstone | synagogue |
| | jerusalem | shooting_in | injured_in |
| | parenthood | gunmen | in_nigeria |

**Table 5: Top features extracted by PMI using unigrams only (top) or unigrams and bigrams (bottom). Results are obtained taking the entire training data for all events (terrorist attacks) as the positive class.**

| | Synthetic training examples (baselines) or seeds (ours) | | | | | |
|---|---|---|---|---|---|---|
| | 100 | 500 | 1,000 | 5,000 | 7,000 | 10,000 |
| **Extracted volume** | | | | | | |
| PMI | 1.7M | 953K | 473K | 290K | 140K | 60K |
| Centroid | 1.0M | 429K | 427K | 196K | 135K | 115K |
| FastText | 2.0M | 664K | 259K | 97K | 116K | 101K |
| KNN | 3.6K | 13.4K | 31.2K | - | - | - |
| Ours - unigrams | 6.2K | 15.1K | 33.5K | 112.2K | 149.8K | 171.2K |
| Ours - synsets | 5.0K | 16.8K | 26.1K | 114.3K | 143.4K | 169.1K |
| **Precision** | | | | | | |
| PMI | 0.005 | 0.005 | 0.01 | 0.020 | 0.050 | 0.100 |
| Centroid | 0.004 | 0.030 | 0.080 | 0.120 | 0.210 | 0.220 |
| FastText | 0.005 | 0.040 | 0.100 | 0.190 | 0.240 | 0.270 |
| KNN | 0.810 | 0.740 | 0.670 | - | - | - |
| Ours - unigrams | **0.880** | 0.760 | **0.690** | 0.570 | 0.540 | **0.460** |
| Ours - synsets | **0.880** | **0.770** | **0.690** | **0.560** | **0.550** | **0.460** |
| **Recall** | | | | | | |
| PMI | 0.409 | 0.601 | 0.621 | 0.623 | 0.627 | 0.629 |
| Centroid | 0.544 | 0.600 | 0.634 | 0.641 | 0.671 | 0.703 |
| FastText | **0.557** | **0.630** | **0.643** | 0.646 | 0.703 | 0.722 |
| KNN | 0.102 | 0.265 | 0.32 | - | - | - |
| Ours - unigrams | 0.090 | 0.269 | 0.348 | 0.682 | 0.745 | 0.787 |
| Ours - synsets | 0.130 | 0.283 | 0.384 | **0.701** | **0.775** | **0.797** |
| **F1 score** | | | | | | |
| PMI | 0.010 | 0.010 | 0.020 | 0.039 | 0.093 | 0.173 |
| Centroid | 0.008 | 0.057 | 0.142 | 0.202 | 0.320 | 0.335 |
| FastText | 0.010 | 0.075 | 0.173 | 0.294 | 0.358 | 0.393 |
| KNN | 0.181 | 0.390 | 0.433 | - | - | - |
| Ours - unigrams | 0.163 | 0.397 | 0.463 | 0.621 | 0.626 | 0.581 |
| Ours - synsets | **0.227** | **0.414** | **0.493** | **0.623** | **0.643** | **0.583** |

**Table 6: Evaluation results for the micropost extraction task of the four baseline methods against our method. The average size of a synset pattern was 204, 373, 465, 439, 451, 462 attributes for 100 - 10,000 training examples respectively.**

**4. k-NN-based on WMD metric.** We use the $k$-nearest neighbors (k-NN) method described in [8]. The distance between each new, unseen element to all training examples is computed using the WMD metric as described in the previous sections.

k-NN requires negative samples in addition to the positive samples. Using as negative examples a sample of documents from the main document corpus will not be helpful in this case. Taking into account the abundance of possible topics in the microblogging space, a small training sample of tweets not related to the target topic cannot guarantee that a topic of a randomly picked document will be present in the training dataset (as this probability will be very small). So, for the majority of documents, all documents from the training dataset are equally far and majority vote provides a nearly random answer.

To avoid this problem and be able to use the k-NN approach (since it is one of the very few methods that can be effective even with small training samples) we modify it so that it can work without negative training samples. The idea is to assign the positive class to the documents that have at least K positive documents from the training seed in their near proximity. We selected K to be equal to 3 (as lower numbers significantly decrease precision) and the radius to 0.5 WMD (according the result that we discussed in the previous section).

## 5.2 Metrics and their estimation

We report standard information retrieval metrics: precision, recall and F-measure. Precision was evaluated using 3 random samples of 200 tweets each, which were labeled by the authors of this paper with annotators agreement of 95%[7], following the procedure described in Section 3.2. Computation of recall is challenging since human annotation of the full corpus of 60M tweets is beyond our resources; thus, we rely on relative recall. Relative recall is computed by taking the union of all microposts that are positively labeled by all methods. We report recall as: $\mathrm{RR}_{\mathrm{method}} = \frac{\mathrm{TP}_{\mathrm{method}}}{\sum_{m \in \mathrm{all\_methods}} \mathrm{TP}_m}$, where RR stands for relative recall, and $\mathrm{TP}_{\mathrm{method}}$ is a true positive rate for a given method. Finally, we report F-measure as follows: $\mathrm{F}_{\mathrm{method}} = \frac{2 * \mathrm{P} * \mathrm{RR}}{\mathrm{P} + \mathrm{RR}}$.

## 5.3 Results

Results are summarized in Table 6. Our method performs better than the baselines in terms of both precision and recall when we allow it to use 5’000 or more automatically selected seeds. Synset-based variation of the attributes also performs better than the baselines in terms of F-measure when we use 100 or more automatically selected seeds. In addition, we compare the results of the micropost extraction task using a less sophisticated approach for the synset generation, e.g. when synsets are generated by using the top-10 most similar words for each of the 30K most frequent words in the dataset. This experiment yields a reduction of 3% and 1% for recall and precision respectively, compared to the results obtained using synsets generated by our method (see textit“Ours - unigrams” and *“Our - synsets”* in Table 6). The baselines perform worse in terms of both precision and recall when the number of positively labeled examples are over 5K; in principle this cannot be attributed to the training set quality, as according to our tests it was 95% precise as discussed in Section 3.2.

Our method, in contrast, loses recall on smaller input sizes but wins precision depending on the number of automatically selected seeds to be used, with the best values of F-measure obtained when using around $k = 5,000 - 7,000$ automatically selected seeds. Overall, we observe that our method with any number of $k \geq 100$ automatically selected seeds outperforms all baselines in terms of F-measure, even in cases where they use 10,000 manually labeled items [8]. Table 7 presents samples of patterns and associated documents that are generated by our method.

## 5.4 Discussion

Our approach is most similar to the nearest neighbors approach (kNN); indeed, the results of both approaches on small trainings sets are comparable. However, our approach does not have the limitations that kNN has:

- Unlike kNN, we do not require objects with negative class labels. Collecting samples of tweets that are not related to the topic is often impractical, as the number of potential topics to cover can be very large.
- Our method is more robust to large training samples (which are potentially more noisy) and complex distance metrics. Word Mover’s Distance has a computational complexity $O(w^3 log(w))$, where $w$ is the average length of a document. Multiplied by the size of a training set $|T|$ and the size of the text corpus $|D|$ makes it impractical for large collections. In our case, we were not able to get results for training sets larger than 1’000 documents for kNN, as the extraction process on a cluster of 50 machines was still running after several days.

**Empirically, topics are mixtures of sub-topics.** Compared to the baselines, our method shows stable performance across all seed sizes. It is noticeably more selective, especially on smaller samples, where the non-kNN methods perform quite poorly. With more seeds, our methods still maintains a high precision and outperforms the baselines in terms of recall. The level of precision is guaranteed by the topical compactness of the extracted patterns, which is a key element of our method. The increase in recall is also expected for higher numbers of seed documents as it allows us to cover more subtopics and consequently more relevant microposts.

One possible reason why the Centroid and FastText approaches do not significantly benefit from growing seed sizes is that they conceptually try to find a clear center in the embedding space that is supposedly the pivot of the topic. This is in contrast to an empirical observation, which shows that each topic is typically a mixture of numerous smaller subtopics that have little overlap between each other. For example, here are several tweets that were considered to be related to terrorist attacks, but that do not have much in common:

(1) “Amnesty International Says Boko Haram Kills Thousands in

[7] Microposts describing an event from the past lead to the most annotation disagreement, since those were not specifically reflecting an event that has recently happened. We included such examples to the training set.

[8] We have also performed a robustness test against noise in the training set (1%, 2%, 5%, 10% of noise). As a result, P and R were equivalent to the results presented in Table 6 for any size of the training set. However, the number of the extracted patterns on average were 20% lower.

| Mean WDM | Pattern | Support | Micropost examples |
|---|---|---|---|
| 0.436 | attack, claim, SYNSET_Egypt | 99 | "isis claims responsibility for tunisia attack that killed 13 people" "islamic state claims responsibility for tunisia attack statement reuters" … |
| 0.476 | boko, SYNSET_attack | 686 | "boko haram gunmen attack nigeria villages kill 43publish date feb 13 2014 new vision #bokoharam" "flash buhari explains legal basis for accepting suvs after boko haram attacked him in kaduna in 2014" "boko haram gunmen attack nigeria villages kill 43" … |
| 0.375 | boko, bomber, femal, haram | 41 | "alleged boko haram suicide bombers dressed as females die in an accident in borno see photos" "see photos of the 13 year old female boko haram suicide bomber" "ttw today s news suspected boko haram female suicide bombers blow up market in nigeria" "breaking boko haram attacks maiduguri again as female suicide bombers did this via" |
| 0.474 | bomber, polic, suicid | 128 | "turkey suicide bomber wounds 5 turkish police during r #trending #news #startups #howto #diy #android #howto #apps" "muslim b tch blows up police dog heroic k9 diesel blown up by female suicide bomber in paris #mcgnews" "french suicide bomber killed during raid was blonde woman yelling help me to police before she detonated bomb bb4sp" "french honor diesel hero police dog blown up by suicide bomber during terrorists last stand #jesuischien" "is says dutch suicide bomber struck iraq police #middleeast #politics" "police suicide bombers of 11 female target nigeria market" |
| 0.345 | attack, government, militant, somali | 53 | "al shabaab militants attack somali government building at least 5 dead mogadishu reuters a #breakingnews" "somali militants raid government base at least eight people are killed in an attack by suspected al shabab mi" "world somali police say 7 dead in attack on baidoa government hq mogadishu somalia suspected islamic militants" |
| 0.399 | attack, blast, kabul | 87 | "blast and gunfire in kabul s diplomatic district second attack in a day fighting season ends in the battlefield begins in kabul" "rt updated story deadly blast at kabul airport as taliban attacks surge" "after blast in kabul taliban say they made suicide attacks against guesthouse for foreigners" "rt after blast in kabul taliban say they made suicide attacks against guesthouse for foreigners" "updated story deadly blast at kabul airport as taliban attacks surge" |

**Table 7: Examples of patterns and associated documents generated by our approach. Mean WDM refers to mean pair-wise WDM document distance. Pattern is presented as a combination of stemmed words and synsets.**

Nigeria's Baga Town …"; (2) "Palestinian Kidnapped Near #Jenin #westbank"; (3) "ISIS releases internet video purportedly showing American journalist Steven Sotloff's beheading"; (4) "Twin suicide bomb blast rocks northern #Cameroon village"; (5) "As usual terrorist attacks take place in Sinai, while military will strike back against university students and women in rest of #Egypt"

PMI adjusts to general words like "attack", "terrorist", "massacre", "killed", etc., which explains the reason why it shows a relatively high recall on training sets of different sizes. This also explains the low precision values: that generality does not allow PMI to discern terrorism from other topics related to casualties, deaths, or violence.

Precision, in our approach, slightly degrades with larger training sets. We attribute this to a growing number of outliers that are included into training samples.

## 6 CONCLUSIONS

In this paper we introduced a generic and flexible framework for semantic filtering of microposts. Our framework processes microposts by combining two key features: semantic pattern mining and document similarity estimation based on the extracted patterns.

Compared to the baselines, our method shows stable performance across all document seed sizes. It is noticeably more selective, especially on smaller samples, where the non-kNN methods perform quite poorly. In particular, our approach leverages word embeddings that are trained on event-specific microposts, thus enhancing the event representation on particular Social Media platforms. Our approach makes no use of external knowledge bases (e.g., WordNet) nor of linguistic tools (parsers) that are computationally expensive. Our empirical results show that our algorithm is efficient and can process high-velocity streams, such as the Twitter stream, in real-time. We demonstrates its efficiency on a large corpus and showed that our topical extraction outperforms state-of-the-art baselines.

**Future Work.** Our current method of topical pattern extraction uses two attributes that represent the documents: stemmed unigrams and synsets. These attributes can be expanded with potentially more expressive features like n-grams, entity types, etc., thus, the method could adapt to finer topical nuances. To make our approach more efficient, we plan to optimize the pattern extraction process even further. The idea is to apply restrictive pattern growing techniques that prevent the emergence of multiple patterns based on similar sets of documents. Finally, as embeddings are usually highly dependant on the input, mixed embeddings (e.g., trained on both Social Media content and Wikipedia) could be leveraged to make our method more robust.

**Data and code availability.** Code and anonymized data are available at https://github.com/toluolll/ShortTextFiltering

**Acknowledgments.** We would like to thank Dr. Vincent Lenders from ArmaSuisse for providing the microposts corpus. J.P. was supported by a Sinergia Grant by the Swiss National Science Foundation (SNF 147609). C.C. worked on this paper while at Eurecat, supported by the Catalonia Trade and Investment Agency (ACCIÓ). P.C.M. was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 683253/GraphInt).

## REFERENCES

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Acm sigmod record*, Vol. 22. ACM, 207–216.

[2] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Gker, I. Kompatsiaris, and A. Jaimes. 2013. Sensing Trending Topics in Twitter. *IEEE Transactions on Multimedia* 15, 6 (Oct 2013), 1268–1282. DOI: http://dx.doi.org/10.1109/TMM.2013.2265080

[3] Mohammad Akbari, Xia Hu, Nie Liqiang, and Tat-Seng Chua. 2016. From Tweets to Wellness: Wellness Event Detection from Twitter Streams. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 87–93.

[4] Nasser Alsaedi, Pete Burnap, and Omer Rana. 2016. Sensing Real-World Events Using Arabic Twitter Posts. (2016).

[5] Gaurav Baruah, Mark D. Smucker, and Charles L.A. Clarke. 2015. Evaluating Streams of Evolving News Events. In *Proc. of Conference on Research and Development in Information Retrieval (SIGIR) (SIGIR '15)*. ACM, New York, NY, USA, 675–684.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). http://arxiv.org/abs/1607.04606

[7] Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless Text Classification with Descriptive LDA. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2224–2231.

[8] T. Cover and P. Hart. 2006. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theor.* 13, 1 (Sept. 2006), 21–27.

[9] Lishan Cui, Xiuzhen Zhang, Xiangmin Zhou, and Flora Salim. 2016. *Topical Event Detection on Twitter*. Springer International Publishing, Cham, 257–268. DOI: http://dx.doi.org/10.1007/978-3-319-46922-5_20

[10] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. 2012. Finding Bursty Topics from Microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 (ACL '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 536–544.

[11] S. Girtelschmid, A. Salfinger, B. Prll, W. Retschitzegger, and W. Schwinger. 2016. Near real-time detection of crisis situations. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 247–252. DOI: http://dx.doi.org/10.1109/MIPRO.2016.7522146

[12] Qi He, Kuiyu Chang, and Ee-Peng Lim. 2007. Analyzing Feature Trajectories for Event Detection. In *Proc. of Conference on Research and Development in Information Retrieval (SIGIR) (SIGIR '07)*. ACM, New York, NY, USA, 207–214.

[13] Weijing Huang, Wei Chen, Lamei Zhang, and Tengjiao Wang. 2016. *An Efficient Online Event Detection Method for Microblogs via User Modeling*. Springer International Publishing, Cham, 329–341.

[14] Shan Jiang, Yuening Hu, Changsung Kang, Tim Daly, Jr., Dawei Yin, Yi Chang, and Chengxiang Zhai. 2016. Learning Query and Document Relevance from a Web-scale Click Graph. In *Proc. of Conference on Research and Development in Information Retrieval (SIGIR) (SIGIR '16)*. ACM, New York, NY, USA, 185–194.

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[16] S. Katragadda, S. Virani, R. Benton, and V. Raghavan. 2016. Detection of event onset using Twitter. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 1539–1546. DOI: http://dx.doi.org/10.1109/IJCNN.2016.7727381

[17] Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 1411–1420.

[18] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings to Document Distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15)*. JMLR.org, 957–966.

[19] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *Proc. of Conference on Research and Development in Information Retrieval (SIGIR) (SIGIR '16)*. ACM, New York, NY, USA, 165–174.

[20] Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective Document Labeling with Very Few Seed Words: A Topic Model Approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 85–94.

[21] Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. 2012. TEDAS: A Twitter-based Event Detection and Analysis System. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering (ICDE '12)*. IEEE Computer Society, Washington, DC, USA, 1273–1276.

[22] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text Classification by Labeling Words. In *Proceedings of the 19th National Conference on Artifical Intelligence (AAAI'04)*. AAAI Press, 425–430.

[23] Xiaomo Liu, Quanzhi Li, Armineh Nourbakhsh, Rui Fang, Merine Thomas, Kajsa Anderson, Russ Kociuba, Mark Vedder, Steven Pomerville, Ramdev Wudali, Robert Martin, John Duprey, Arun Vachher, William Keenan, and Sameena Shah. 2016. Reuters Tracer: A Large Scale System of Detecting and Verifying Real-Time News Events from Twitter. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 207–216.

[24] Jonathan Matusitz. 2012. *Terrorism and communication: A critical introduction*. Sage Publications.

[25] R. McCreadie, C. Macdonald, I. Ounis, M. Osborne, and S. Petrovic. 2013. Scalable distributed event detection for Twitter. In *2013 IEEE International Conference on Big Data*. 543–549. DOI: http://dx.doi.org/10.1109/BigData.2013.6691620

[26] Duc T. Nguyen and Jason J. Jung. 2015. Real-time Event Detection on Social Data Stream. *Mob. Netw. Appl.* 20, 4 (Aug. 2015), 475–486.

[27] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2014. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises *(ICWSM '14)*.

[28] Nikolaos Panagiotou, Ioannis Katakis, and Dimitrios Gunopulos. 2016. *Detecting Events in Online Social Networks: Definitions, Trends and Challenges*. Springer International Publishing, Cham, 42–84. DOI: http://dx.doi.org/10.1007/978-3-319-41706-6_2

[29] Min Peng, Jiahui Zhu, Xuhui Li, Jiajia Huang, Hua Wang, and Yanchun Zhang. 2015. Central Topic Model for Event-oriented Topics Mining in Microblog Stream. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 1611–1620.

[30] Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming First Story Detection with Application to Twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 181–189.

[31] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open Domain Event Extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 1104–1112.

[32] Bella Robinson, Robert Power, and Mark Cameron. 2013. A Sensitive Twitter Earthquake Detector. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13 Companion)*. ACM, New York, NY, USA, 999–1002.

[33] Yu Rong, Qiankun Zhu, and Hong Cheng. 2016. A Model-Free Approach to Infer the Diffusion Network from Event Cascade. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 1653–1662.

[34] Mehdi Sadri, Sharad Mehrotra, and Yaming Yu. 2016. Online Adaptive Topic Focused Tweet Acquisition. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 2353–2358.

[35] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 851–860.

[36] Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction. In *Proceedings of CoNLL 2015*.

[37] Yangqiu Song and Dan Roth. 2014. On Dataless Hierarchical Text Classification. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 1579–1585.

[38] Alberto Tonon, Philippe Cudré-Mauroux, Albert Blarer, Vincent Lenders, and Boris Motik. 2017. ArmaTweet: Detecting Events by Semantic Tweet Analysis. In *ESWC*. 138–153. DOI: http://dx.doi.org/10.1007/978-3-319-58451-5_10

[39] Zeynep Tufekci. 2017. *Twitter and Tear Gas: The Power and Fragility of Networked Protest*. Yale University Press.

[40] Maximilian Walther and Michael Kaisser. 2013. Geo-spatial Event Detection in the Twitter Stream. In *Proceedings of the 35th European Conference on Advances in Information Retrieval (ECIR'13)*. Springer-Verlag, Berlin, Heidelberg, 356–367.

[41] Chao Wang, Xue Zhao, Ying Zhang, and Xiaojie Yuan. 2016. *Online Hot Topic Detection from Web News Based on Bursty Term Identification*. Springer International Publishing, Cham, 393–397.

[42] Mohammed Javeed Zaki, Srinivasan Parthasarathy, and Wei Li. 1997. A localized algorithm for parallel association mining. In *Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*. ACM, 321–330.

[43] Hamed Zamani and W. Bruce Croft. 2016. Estimating Embedding Vectors for Queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR '16)*. ACM, New York, NY, USA, 123–132.

[44] Shanshan Zhang and Slobodan Vucetic. 2016. Semi-supervised Discovery of Informative Tweets During the Emerging Disasters. *CoRR* abs/1610.03750 (2016).