Event Detection for Heterogeneous News Streams

Ida Mele and Fabio Crestani

Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland
{ida.mele, fabio.crestani}@usi.ch

Abstract. In this paper we tackle the problem of detecting events from multiple and heterogeneous streams of news. In particular, we focus on news which are heterogeneous in length and writing styles since they are published on different platforms (i.e., Twitter, RSS portals, and news websites). This heterogeneity makes the event detection task more challenging, hence we propose an approach able to cope with heterogeneous streams of news. Our technique combines topic modeling, named-entity recognition, and temporal analysis to effectively detect events from news streams. The experimental results confirmed that our approach is able to better detect events than other state-of-the-art techniques and to divide the news in high-precision clusters based on the events they describe.

Keywords: event detection, news clustering, heterogeneous news streams

1 Introduction

Topic Detection and Tracking (TDT) is an important research area which has attracted a lot of attention especially in information analysis for discovering newsworthy stories and studying their evolution over time. The main challenge of TDT is to group news that are about some specific events (e.g., Ecuador earthquake) and tracking their evolution over time. In this paper, we focus on the problem of event detection from multiple and heterogeneous streams of news, namely, news reported by different channels (e.g., BBC, CNN) on different publishing platforms (e.g., Twitter, RSS portals, and news websites)[1].

In the past, event detection has received a lot of attention, but most of the approaches presented in the literature focus on one stream of text or tackle the problem of event detection in news articles and in tweets, separately [4,6,7,15]. This could be a limitation when we deal with heterogeneous news coming from multiple streams, since techniques that are effective for long text (e.g., news articles) are known to give poor performance when applied to short text (e.g., tweets). On the other hand, considering event detection from multiple streams as a single task, which is irrespective of the document type, can take advantage of cross-linking the news.

[1] The words *channel* and *stream* are used interchangeably in this paper, and we use the general term *news* to refer to a news article, an RSS feed, or a tweet

To address the problem of TDT in multiple and heterogeneous streams, we developed an approach which leverages topic models, named-entity recognition, and temporal analysis of bursty features. As we will see in Section 5, our approach overcomes traditional techniques for event detection [4,15] and document clustering [3]. It captures crisp events, divides the news in high-precision clusters and is document independent in the sense that it can be used for different types of news (e.g., short tweets as well as long news articles). It is also query-less which means that we do not need predefined queries and this is beneficial especially for independently discovering emerging topics or for analyzing events for which we have limited background knowledge. In addition, our approach relies on features, such as named entities and event phrases, providing a short but understandable description of the event. Finally, the size of an event's window is automatically mined from the data.

The contributions of this paper are the following:

1. we present an approach which detects events by applying topic mining, named-entity recognition, and temporal analysis of bursty features on news;
2. we cluster the news based on the events they describe;
3. we compare different methodologies for event detection and text clustering focusing on TDT in multiple and heterogeneous streams of news.

The rest of the paper is structured as follows: we review related work in Section 2. Section 3 describes our methodology for event detection and news clustering. We discuss the characteristics of the approach in Section 4 and present the experimental results in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Topic Detection and Tracking (TDT) is a wide research area which includes several tasks ranging from event segmentation of news streams to event detection and tracking. Event detection is based on monitoring streams of news and automatically organizing the news by the events they describe. Research works on event detection can be divided into two categories: *document-pivot* and *feature-pivot* approaches. The former focuses on clustering documents related to the same event and then extracting the event-based features from the discovered clusters [1]. The latter is based on finding hidden features and then clustering these features in order to identify the events from the news [4,6,15].

Fung et al. [4] addressed the problem of detecting *hot bursty event features*. Their technique finds a minimal set of features representing the events in a specific time window. They first identify bursty features by statistically modeling the frequency of each unigram with a binomial distribution. Then, they group these features into events and use time series analysis to determine the hot period of an event. The extraction of bursty features based on statistics may result in a prohibitive number of features, especially when unigrams are used. Moreover, describing the detected events using a set of single words may be not intuitive and difficult for human interpretation.

He et al. [6] treat signals as features and apply Discrete Fourier Transformation (DFT) on them. Their approach builds a signal for each feature using the *document frequency - inverse document frequency* (df × idf) scheme along with the time domain. Then, it applies DFT to transform the signal from the time domain to the frequency domain. A spike in the frequency domain indicates a corresponding high-frequency signal source. Such bursty features are then grouped into events by considering both the features' co-occurrences and their distributions in the time domain. This approach has scalability issues due to the application of DFT which can be computationally prohibitive.

In the last few years, research works focused on detecting events in Twitter. The challenge is that traditional approaches developed for formal text (e.g., news articles) cannot be applied directly to tweets, which are short, noisy, and published at a high-speed rate. One of the main problems in microblogging systems is to distinguish the newsworthy events from trends or mundane events which attract attention from fans and enthusiasts. In [2] the authors present an approach for separating real-world events from non-event tweets based on aggregated statistics of temporal, topical, social, and Twitter-centric features. Another approach consists in finding the hashtag *#breakingnews* to identify news in Twitter [10]. In our research, we do not consider these techniques since we monitor news channels, hence their tweets are *all* newsworthy. We rather analyze the streams of tweets to detect the events and divide the tweets into clusters based on the events they describe. A similar problem was addressed by [14] for detecting crime or disasters from tweets. Analogously, Popescu et al. [11] used an entity-based approach for event detection where a set of tweets containing a target entity are processed and machine learning techniques are applied to predict whether the tweets constitute an event regarding the entity or not. The drawback of these approaches is that the events must be known a priori and be easily represented by well-defined keyword queries (e.g., "earthquake") or named entities (e.g., "Obama"). Ritter et al. [13] tried to overcome this limitation by designing an open-domain system for extracting a calendar of categorized events.

Other popular approaches for event detection in Twitter are: *Twevent* [7] and Event Detection with Clustering of Wavelet-based signals (*EDCoW*) [15]. *Twevent* clusters tweets representing events and provides a semantically meaningful description of the clusters. It segments the tweets relying on statistics from Microsoft Web N-Gram service and Wikipedia [8]. Then, bursty segments are detected by analyzing the tweet frequency and user frequency. The assumption is that if a segment is related to an event, then it is present in many tweets which are posted by many different users. *EDCoW* applies measurements to see how signals (unigrams) change over time and uses wavelet analysis to spot high-energy signals which are then treated as event features.

In this paper we do not focus on just one type of document (e.g., tweets), we rather propose a technique for event detection from streams of heterogeneous documents. In Section 5 we will compare our approach against [4] and [15] since they are general and do not need any predefined queries. To the best of our knowledge this is the first work that tackles the problem of event detection from multiple and heterogeneous streams of news. Other works have analyzed multiple news streams but for other purposes, e.g., analyzing the newswires' timeliness [5].

3 *EDNC*: Event Detection and News Clustering

In this section we describe our approach which is called Event Detection and News Clustering (*EDNC*). It allows to detect events from multiple and heterogeneous news streams and to divide the news into corresponding event clusters.

We assume to have n streams of news $N = \{N_1, N_2, ..., N_n\}$, where $N_i = \{n_{i,1}, n_{i,2}, ..., n_{i,m}\}$ is the stream i consisting of m news. We want to analyze the news in order to identify a set of popular events, $E = \{e_1, e_2, ...\}$ that appear in them. Each event, $e_j \in E$, is represented by $\langle w_{e_j}, F_{e_j} \rangle$ where w_{e_j} is the time window of the event and F_{e_j} are variable-length phrases, called *event features*, which give crisp information about the event. In Section 5 we will provide some examples of event features. *EDNC*'s steps are summarized in Figure 1.

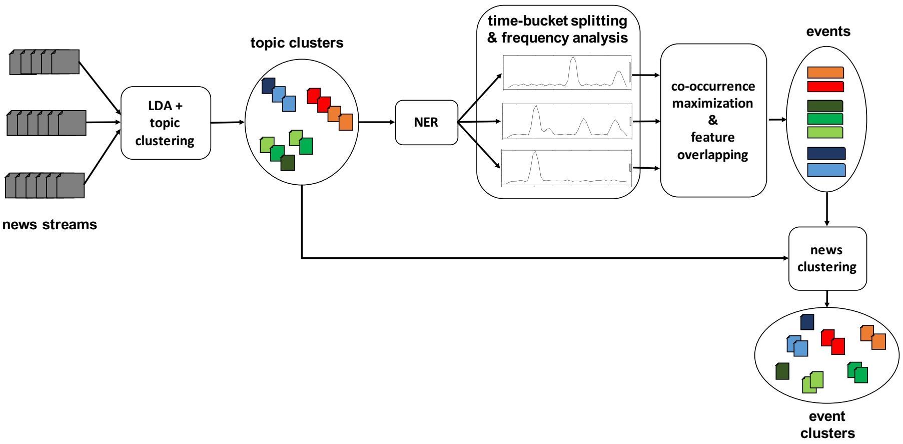

Fig. 1. A diagram showing the main steps of *EDNC* approach.

3.1 Event Detection

Our event-detection methodology first applies LDA to detect general topics and to create topic clusters of news. News are firstly divided into broad topic clusters by applying the approach described in [9], which treats each LDA topic as a cluster. In particular, a document is interpreted as a distribution vector of topics, $\boldsymbol{\theta}$, and it is assigned to the cluster x if $x = argmax_j(\theta_j)$. Although assigning the news to one topic may seem a limitation, it is actually a reasonable solution especially if we think that news are usually about one specific event. Moreover, other studies on topic modeling proved that this approach is effective especially for short text (i.e., RSS feeds and tweets) [16].

As second step, the approach analyzes the frequency of named entities and other representative event phrases over time. To do this, we apply a *named-entity recognition* tool [12] which finds out the named entities and event phrases in a document collection. Then, the news streams are sorted by time and divided into time buckets of fixed size (e.g., 24 hours). *EDNC* computes the frequencies of detected named entities and event phrases in each time bucket. The most frequent named entities and event phrases (e.g., the top-10%) are retained as *event features*, $F_{e_j} = \{f_1, f_2, ...\}$. Note that these are variable-length sequences of words (e.g., "Barack Obama," "tropical storm Colin," "earth shakes") and not just single words. Since these features are very frequent in a time bucket, they can be used to semantically describe the popular events.

3.2 Estimating the Temporal Windows of Events

Maximizing the co-occurrences of the event features, we can divide the events and determine their time span. As an example, in the time bucket [April 17, 2016] if we observe $F = \{Japan, Ecuador, earthquake\}$ as frequent features and there is a high co-occurrence for $\{Ecuador, earthquake\}$ and $\{Japan, earthquake\}$, we can assume that around mid April two earthquakes occurred, one in South America and another one in Japan. Hence, in that time bucket we have two events whose corresponding event features are: $F_{e_1} = \{Ecuador, earthquake\}$ and $F_{e_2} = \{Japan,\ earthquake\}$.

Once the events are identified, we need to determine the size of their temporal windows. They can be different depending on the popularity of the events. For example, news about popular events span over a large period of time (e.g., Brexit) while those related to minor events attract attention only for a limited period of time (e.g., small earthquakes in California). Our approach creates sliding windows over the time buckets and computes the overlapping of the event features in consecutive time buckets. For example, if in the temporal bucket [April 14, 2016] we have $\{Japan,\ Kumamoto,\ earthquake\}$ and in [April 15, 2016] we have $\{Kumamoto,\ earthquake,\ victims\}$, *EDNC* merges these two events since there is overlapping of "Kumamoto" and "earthquake." Such overlapping depends on a parameter which specifies the percentage of shared keywords in the event features, and it can be tuned to get more defined events. In particular, bigger values of the feature-overlapping percentage (e.g., 70%) are used for identifying crisp events. *EDNC* uses consecutive buckets because if two time slots are characterized by similar keywords but they are distant (i.e., the buckets in the middle do not present any of the monitored event features), they should be considered as separated events. For example, if we observe in the time buckets [March 11, 2016] and [April 14, 2016] event keywords like $\{Japan,\ tsunami,\ earthquake\}$ and $\{Japan,\ Kumamoto,\ earthquake\}$, we can assume that they are two distinct events. Indeed, the former is about the 5th anniversary of the tsunami which devastated Japan in March 2011, while the latter is about the earthquake which hit the South of Japan in April 2016.

3.3 Event-Based Clustering of News

We created clusters of news based on the events they describe. Each cluster is identified by the event features and the temporal window.

For creating the clusters we used a traditional IR approach which retrieves the news and rank them based on their similarity to the event features. We first filter out all the news that are not within the event's temporal window, then the news are sorted by their content similarity with the event's features. In particular, we applied the cosine similarity between the news N and the list of keywords in the event features F:

$$\cos(\mathbf{N},\mathbf{F}) = \frac{\mathbf{N}\cdot\mathbf{F}}{\|\mathbf{N}\|\|\mathbf{F}\|} = \frac{\sum\limits_{i=1}^{n} N_i\ F_i}{\sqrt{\sum\limits_{i=1}^{n} N_i^2}\sqrt{\sum\limits_{i=1}^{n} F_i^2}}$$

where $\mathbf{N}$ and $\mathbf{F}$ are two vectors representing the news and the event features, respectively.

4 Characteristics of *EDNC*

As explained in Section 1 we aim at detecting events from multiple and heterogeneous streams of news, and, for this reason, our event-detection tool fulfills the following desiderata:

1. **Document independent.** We focus on different publishing platforms, so the approach works for different types of documents (news articles, RSS feeds, and tweets), while the approaches presented in [7,10] are suitable only for tweets. In particular, they use hashtags as well as retweet popularity (i.e., the number of times the users (re)tweet the news) and such information is not available for news articles and RSS feeds.
2. **Query-less.** We assume that professional users (e.g., journalists and news analysts) would like to *discover* the events without any or limited knowledge of what is going on in the world. Hence, our technique does not need input keywords to retrieve relevant news, while other state-of-the-art approaches detect the events that match some predefined search keywords or named entities [8,11].
3. **Flexible time windows.** Having time windows with variable size allows to cluster together news about popular events which tend to span over a large period of time (e.g., earthquakes, Brexit) as well as minor events which attract attention only for a limited period of time (e.g., 5th anniversary of tsunami in Japan).
4. **Semantically significant description of events.** Our approach can be used to semantically describe the identified events, hence to label the news clusters. The event descriptions are variable-length lists of words (e.g., named entities and event phrases), making the understanding of the event easy even with limited background knowledge about it.

5 Experimental Results

In this section we present the experimental setup and results. In order to evaluate the effectiveness of our approach we collected data from different newswires and different platforms to create a heterogeneous dataset of news documents. We then used this collection for event detection and for clustering the news based on the detected events.

5.1 Dataset

For our experiments we created a dataset of heterogeneous news published by different newswires on different platforms. In particular, we collected news articles, RSS feeds, and tweets published by 9 newswire channels (ABC, Al Jazeera, BBC, CBC, CNN, NBC, Reuters, United Press International, and Xinhua China Agency) for several months. For the experiments reported in this paper, we used the English news published during 4 months (from March 1 to June 30, 2016), for a total of around 140K news documents. Each news document has a title (optional), content, link (optional), timestamp, and channel. The optional fields are present in news articles but may be not available for tweets and RSS feeds.

Since some of the techniques used for event detection are not time efficient, analyzing the whole dataset would be time consuming. So, we subsampled the dataset by selecting 10 topics which were related to some important events that happened in the 4 months of our data, such as terror attacks, Brexit, and earthquakes. News relevant to these topics form broad clusters (i.e., the news are about different, although related, events). For example, the cluster corresponding to the topic *terror attacks* includes several events, such as a bomb at an airport checkpoint in Somalia, shootings at hotels in the Ivory Coast, bomb explosions in Belgium, etc. Some of them can be also connected and interleaved. Consider, for example, the terror attacks that have recently happened in Europe. At the beginning of March, some suspects were arrested in Belgium, followed by suicide bombings in Brussels on March 22, 2016, then at the end of April one of the terrorists was handed over French authorities. Our technique aims at distinguishing these low-granularity events from the topic clusters to create the corresponding smaller event-based clusters of news.

5.2 Event Detection Evaluation

We now describe the methodology we used for the evaluation of the event detection task and the corresponding experimental results.

Evaluation Methodology. We applied *EDNC* for discovering the events reported in the news documents. To evaluate the effectiveness of our approach, we also considered two alternative approaches for event detection which are based on co-occurrences of unigrams (*Unigram Co-Occurrences*) [4] and on wavelet analysis (*EDCoW*) [15]. The former detects the events by analyzing the co-occurrences of bursty features (unigrams) in non-overlapping time windows. The latter is

Table 1. Some of the events detected by the *EDNC* approach in the period of time from March 1 to June 30, 2016. For each event we report the event features plus a short description and the indicative date of the event.

Event Features	Description and Indicative Date
rome, pell, cardinal, cover, abuse, denies	Cardinal Pell's testimony at the child-abuse commission (Mar. 1)
indonesia, quake-strikes, sumatra, tsunami	Earthquake in Sumatra caused tsunami warning in Indonesia (Mar. 2)
anniversary, remember, tsunami, japan, 5-years-ago	Japan marked the 5th anniversary of the 2011 tsunami (Mar. 11)
brussels, captured, paris, salah-abdeslam	Police arrested one of the Paris' terror attacker (Mar. 18)
castro, cuba, havana, obama, visit	Obama visited Cuba (Mar. 21)
attack, brussels-airport, isis, maelbeek-metro	Terror attacks happened in Brussels (Mar. 22)
argentina, mauricio-macri, obama, tango	Obama visited Mauricio Macri in Argentina (Mar. 23)
cyprus, egyptair, hijacking, surrendered	An Egyptair flight was diverted to Cyprus (Mar. 29)
easter, pope, ritual, washed	Pope celebrated the Catholic Easter (Mar. 25)
referendum, dutch, ukraine, eu	Dutch referendum on the Ukraine-EU Association Agreement (Apr. 6)
japan, kumamoto, earthquake, damage, victims	Earthquake hit Kumamoto province in Japan (Apr. 14)
earthquake, ecuador, strikes	Earthquake devastated Ecuador (Apr. 16)
nairobi, kenya, building, collapses, death	A building collapsed in Nairobi, Kenya (Apr. 30)
miami, first, cruise, passengers, cuba	New cruise set sails from Miami to Havana (Apr. 30)
canada, alberta, wildfire, fort-mcmurray, fire	Fort McMurray was evacuated due to wildfires in Alberta (May 4)
crash, disappears, egyptair, flight-ms804	Egyptair plane crashed into the Mediterranean Sea (May 19)
boxing, died, louisville, muhammad-ali	The boxing champion Muhammad Ali died (Jun. 3)
funeral, memorial, muhammad-ali, remembered	Funeral of Muhammad Ali (Jun. 10)
christina-grimmie, singer, voice, orlando, shot	The singer Christina Grimmie was shot during her concert (Jun. 10)
nightclub, orlando, shooting, victims	Several people were killed at a nightclub in Orlando (Jun. 13)
britain, brexit, european, leave-vote, referendum	Brexit referendum in UK (Jun. 23)
virginia, floods, killed, homeland-security, devastating	Flooding in Virginia caused by the heavy rain (Jun. 23)

called Event Detection with Clustering of Wavelet-based signals (*EDCoW*). It creates a signal for each individual word, then it applies *wavelet transformation* and *auto-correlation* to measure the bursty "energy" of the words. Words with high energies are retained as event features and using *cross-correlation* the similarity between pairs of events is measured. Event detection is done by creating a graph consisting of words with high cross-correlation and partitioning it based on the modularity. Both approaches have fixed time windows whose length must be specified as a parameter of the program. We tried different values from 3 to 10 days, and we could observe better results with time windows of 7 days.

Results. Some of the events identified by our methodology are reported in Table 1. We analyzed the corresponding news to provide a short description and help the reader to understand the event features. As we can see, our methodology was able to spot popular events such as Brexit, Obama's visit to Cuba, terror attacks in Brussels, and earthquakes in Asia and South America. It could also detect some minor events such as the hijacking of an Egyptian aircraft and wildfires in Canada.

Table 2 shows the events detected by the other two state-of-the-art approaches (*Unigram Co-Occurrences* and *EDCoW*) with windows of 7 days. As we can see, *Unigram Co-Occurrences* detected more events compared to *EDCoW* which found no events in some of the time windows. Both approaches have two main drawbacks: (1) the event keywords are oftentimes general and difficult to interpret without a manual inspection of the news in the dataset; (2) the size of the temporal window is fixed and must be estimated up front. Moreover, in *EDCoW* the computation of wavelet transformation and auto-correlation is computationally complex and time consuming. Using cross-correlation as similarity measure can result in noisy grouping of events that may have happened

Table 2. Events detected by *Unigram Co-Occurrences* and *EDCoW* approaches using time windows of 7 days, in the period of time from March 1 to June 30, 2016.

Time Window	Unigram Co-Occurrences	EDCoW
w_0 Mar. 1–7	pell, abuse tsunami, quake	cuba, damage, killed, told, vatican
w_1 Mar. 8–14	louisiana, flooding	cuba, left, meet louisiana, rain, white house
w_2 Mar. 15–21	abdeslam, paris obama, cuba	–
w_3 Mar. 22–28	brussels, attacks police, abaaoud	meet, obama, visit francis, pope
w_4 Mar. 29–Apr. 4	plane, hijacker	–
w_5 Apr. 5–11	ukraine, dutch	–
w_6 Apr. 12–18	quake, japan ecuador, earthquake	affected, left, reported, struck damage, death toll, hit, missing, working
w_7 Apr. 19–25	obama, british	–
w_8 Apr. 26–May 2	cuba, cruise building, kenya	–
w_9 May 3–9	mcmurray, fire tornado, oklahoma	fire, fort-mcmurray, started
w_{10} May 10–16	lightning, bangladesh	fire, flooding, rain, reported
w_{11} May 17–23	flight, egyptair everest, summit	–
w_{12} May 24–30	vietnam, obama	–
w_{13} May 31–Jun. 6	boxing, ali germany, lightning	funeral, kentucky, muhammad-ali, vietnam
w_{14} Jun. 7–13	grimmie, christina ali, service	–
w_{15} Jun. 14–20	orlando, mateen	gun control
w_{16} Jun. 21–27	britain, brexit	–
w_{17} Jun. 28–30	virginia, emergency	–

in the same period of time just by chance. For example, in w_0: "cuba, damage, and vatican" are grouped together but they refer to different events. In particular, "cuba" probably refers to the news on preparations in Cuba for Obama's future visit, "damage" to the damages caused by the earthquake in Indonesia, and "vatican" to the scandal that involved some Catholic priests in Australia.

5.3 News Clustering

News clustering consists in dividing the news based on the event they report. We now present the evaluation methodology and the experimental results obtained for news clustering.

Evaluation Methodology. Once we detected the events in the collection, we divided the news into event-based clusters. To do so, our methodology computes the cosine similarity between the news documents and the event features as explained in Section 3.3.

Since we aim at capturing news stories reporting the same event, it is crucial to have clusters whose news are truly related to the event. So we focused on

Table 3. Examples of news clustered by *EDNC* based on the events.

Event Features	Time Window	News/Tweets
anniversary, remember, tsunami, japan, 5-years-ago	[Mar. 09 – 11]	Mar. 11: Japan marks fifth tsunami anniversary Mar. 11: Five years ago giant earthquake tsunami hit northeast Japan Mar. 11: Remembering Japan's 2011 tsunami disaster
castro, cuba, havana, obama, visit	[Mar. 18 – 26]	Mar. 20: Obama heads to Havana for historic visit Mar. 20: Barack Obama's visit to Cuba raises hopes Mar. 20: #Cuba to welcome Obama
argentina, mauricio-macri, obama, tango	[Mar. 21 – 27]	Mar. 23: Obama meets Argentine leader Mar. 23: Obama and family arrive in Argentina Mar. 24: Watch the Obamas dance the tango in Argentina
cyprus, egyptair, hijacking, surrendered	[Mar. 29 – 30]	Mar. 29: EgyptAir Jet Hijacked, Diverted to Cyprus Mar. 29: Hijacker forces EgyptAir flight to land in Cyprus Mar. 30: 'What should one do?'- #EgyptAir hijacker
japan, kumamoto, earthquake, damage, victims	[Apr. 14 – 21]	Apr. 16: At least 24 killed after Japan jolted by pair of deadly earthquakes Apr. 16: Consecutive, deadly earthquakes rock southern Japan Apr. 17: Japan quakes: Dozens killed; rescue efforts hampered
earthquake, ecuador, strikes	[Apr. 17 – 26]	Apr. 17: An #earthquake jolts #Ecuador's Pedernales Apr. 17: Strong quake hits off coast of Ecuador, tsunami waves possible Apr. 18: #Ecuador quake toll likely to rise 'in a considerable way'

the precision of the clusters rather than the recall. The precision is defined as the number of news that are relevant to the event over the number of news in the event cluster, where *relevant* means that the news content is about the event. For annotating the news with respect to their relevance to an event, we randomly selected some of the events and used CrowdFlower[2] to collect labels on the relevance of the news to the events. For each news-event pair we collected 3 judgements, and to ensure high-quality evaluation, we removed those news-event pairs for which the evaluators' confidence was less than 2/3.

We compared the news clustering obtained with our methodology against *k-means* [3] and a temporal-aware version of it. The unsupervised clustering algorithm, *k-means*, applies cosine similarity to find similar news and group them into clusters. We run *k-means* with different values of k and observed a good trade-off between precisions and cluster sizes with $k = 500$. The resulting clusters do not have any label describing the news in them, so we had to manually check the content of the clusters to figure out the corresponding events and create a short description of it to show together with the news to the CrowdFlower's evaluators. Since *k-means* per se is unaware of the timestamps of the news documents, we implemented the *k-means+time* approach which applies *k-means* and then filters out news that are not within the temporal window of the event.

Results. Examples of news clusters obtained with our approach are shown in Table 3. For each event we also report the time window. We can notice that the time-window length can vary depending on the events and some of them can span for long periods of time (e.g., earthquakes were popular for about 10 days).

Table 4 shows the average precisions achieved by the different clustering approaches. Results show that *k-means* has low precisions compared to our approach. In particular, it tends to group similar events that happened in different time windows. For example, it does not separate the news about the Japan's recent earthquake from the ones about the tsunami's 5th anniversary or the

[2] `https://www.crowdflower.com/`

Table 4. Average precision of the event clusters obtained with different methodologies.

	k-means	**k-means +time**	**EDNC**
Avg. Precision	0.51	0.83	**0.93**

wildfires in California from the one in Canada. To filter out these noisy news, we implemented the clustering techniques called *k-means+time* which removes the news whose timestamps do not belong to the time intervals of interest. We could observe that when the time windows are taken into account the precision improves, but still there are more false positives compared to our approach.

On the other hand, *EDNC* has a low number of false positives and, consequently, a good value of precision. We noticed that false positives are caused by less popular events and overlapping of event features. For example, news like "Cuban concerns over Venezuela's economic woes..." and "Cuba's combat rappers fight for the country's youth..." were wrongly grouped with the news about Obama's visit to Cuba. This was probably due to the fact that less popular events (e.g., represented by few news in the dataset) are not captured, hence their news are clustered together with the next most similar ones. Other news about Ecuador earthquake were clustered with the news on Japan earthquake, and the manual assessors considered these as false positives. Inspecting the data, we could notice that in these news articles there is the word "Japan" because the two earthquakes happened in the same days and there were discussions on the possibility that they were somehow connected. *EDNC* considered these news relevant to both earthquakes because of the keywords "Ecuador" and "Japan."

6 Conclusions and Future Work

In this paper we propose a technique for event detection and news clustering. Our approach extracts real-world events from heterogeneous news streams and divides the news based on the events they report.

As future work, we would like to analyze the evolution of the events and how they are connected (e.g., the immigration crisis followed by the Pope speech about welcoming immigrants looking for asylum, or the slaughter in Orlando followed by the Obama's visit to the families of the victims). We also plan to explore other applications of this methodology, such as news summarization.

Acknowledgement

This research was partially funded by the the Secrétariat d'Etat à la formation, à la recherche et à l'innovation (SEFRI) under the project AHTOM (Asynchronous and Heterogeneous TOpic Mining).

References

1. Allan, J., Papka, R., Lavrenko, V.: On-line New Event Detection and Tracking. In: 21st International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 37–45. ACM, New York, NY, USA (1998)
2. Becker, H., Naaman, M., Gravano, L.: Beyond Trending Topics: Real-World Event Identification on Twitter. In: International AAAI Conference on Web and Social Media (2011)
3. Ding, C., He, X.: K-means Clustering via Principal Component Analysis. In: 21st International Conference on Machine Learning. pp. 29–38. ACM, New York, NY, USA (2004)
4. Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.: Parameter Free Bursty Events Detection in Text Streams. In: 31st International Conference on Very Large Data Bases. pp. 181–192. VLDB Endowment (2005)
5. Gwadera, R., Crestani, F.: Mining and Ranking Streams of News Stories Using Cross-stream Sequential Patterns. In: 18th ACM Conference on Information and Knowledge Management. pp. 1709–1712. ACM, New York, NY, USA (2009)
6. He, Q., Chang, K., Lim, E.P.: Analyzing Feature Trajectories for Event Detection. In: 30th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 207–214. ACM, New York, NY, USA (2007)
7. Li, C., Sun, A., Datta, A.: Twevent: Segment-based Event Detection from Tweets. In: 21st ACM International Conference on Information and Knowledge Management. pp. 155–164. ACM, New York, NY, USA (2012)
8. Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B.S.: TwiNER: Named Entity Recognition in Targeted Twitter Stream. In: 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 721–730. ACM, New York, NY, USA (2012)
9. Lu, Y., Mei, Q., Zhai, C.: Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. Inf. Retr. 14(2), 178–203 (2011)
10. Phuvipadawat, S., Murata, T.: Breaking News Detection and Tracking in Twitter. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03. pp. 120–123. IEEE Computer Society, Washington, DC, USA (2010)
11. Popescu, A.M., Pennacchiotti, M., Paranjpe, D.: Extracting Events and Event Descriptions from Twitter. In: 20th International Conference Companion on World Wide Web. pp. 105–106. ACM, New York, NY, USA (2011)
12. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named Entity Recognition in Tweets: An Experimental Study. In: Conference on Empirical Methods in Natural Language Processing. pp. 1524–1534. Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
13. Ritter, A., Mausam, Etzioni, O., Clark, S.: Open Domain Event Extraction from Twitter. In: 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1104–1112. ACM, New York, NY, USA (2012)
14. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: 19th International Conference Companion on World Wide Web. pp. 851–860. ACM, New York, NY, USA (2010)
15. Weng, J., Lee, B.S.: Event Detection in Twitter. In: International AAAI Conference on Web and Social Media (2011)
16. Yan, X., Guo, J., Lan, Y., Cheng, X.: A Biterm Topic Model for Short Texts. In: 22nd International Conference Companion on World Wide Web. pp. 1445–1456. ACM, New York, NY, USA (2013)