



Good and bad events: combining network-based event detection with sentiment analysis

Iraklis Moutidis¹ · Hywel T. P. Williams¹

Received: 29 February 2020 / Revised: 23 July 2020 / Accepted: 27 July 2020 / Published online: 5 August 2020
© The Author(s) 2020

Abstract

The huge volume and velocity of media content published on the Web presents a substantial challenge to human analysts. In prior work, we developed a system (network event detection, NED) to assist analysts by detecting events within high-volume news streams in real time. NED can process a heterogeneous stream of news articles or social media user posts, combining text mining and network analysis to detect breaking news stories and generate an easy-to-understand event summary. In this paper, we expand the NED event detection and summarisation approach in two ways. First, we introduce a new approach to named entity disambiguation for tweets, which contain minimal information due to brevity. Second, we apply sentiment analysis techniques to documents associated with a detected event to characterise the event as either broadly ‘positive’ or ‘negative’ based on media portrayal. Our expansion focuses on Twitter streams since Twitter has become an important news dissemination platform and is often the site where emerging events are first seen. To test the extended methodology, we apply it here to three data sets related to political elections in the UK and the USA. The addition of sentiment analysis to the NED event detection methodology improves the insight gained by the user by allowing quick evaluation of the perceived impact of an event. This approach may have potential applications in domains where public sentiment is relevant to decision-making around events, such as financial markets and politics.

Keywords Natural language processing · Network analysis · Sentiment analysis · Social media · Topic modelling

Abbreviations

NED Network event detection
TDT Topic detection and tracking
NER Named entity recognition

1 Introduction

The growth and spread of the World Wide Web has radically changed the way news content is published and disseminated to the public. Traditional news platforms have moved online, while user-generated content such as blogs and social media have expanded the variety and availability of media

content. This ongoing transformation has many advantages, but also creates new challenges for human analysts whose role requires comprehension of new information, e.g. for trading decisions in financial markets, insurance risk estimation or open source intelligence applications. The volume and velocity of the media ecosystem have become so big that human analysts cannot manually process and make sense of all the information that may be relevant. This creates the need for automated systems that can assist human analysts by processing large volumes of content from heterogeneous news streams in real time, detect and track breaking news events, and provide informative summarisations of complex media data sets. Such information can help users to make timely and well-informed decisions.

In prior work Moutidis and Williams (2019), we introduced a method labelled the network event detection (NED) system (Fig. 1). NED is able to process heterogeneous streams of textual news documents, such as news articles, social media posts (Twitter, Reddit) and blog posts, identify important named entities (People, Organisations) within the evolving news stream, detect breaking news stories and generate an informative summary of each news event. The

Iraklis Moutidis and Hywel T. P. Williams have contributed equally to this work.

✉ Iraklis Moutidis
imoutidi@gmail.com

Hywel T. P. Williams
h.t.p.williams@exeter.ac.uk

¹ University of Exeter, North Park Road, Exeter EX4 4QE, UK

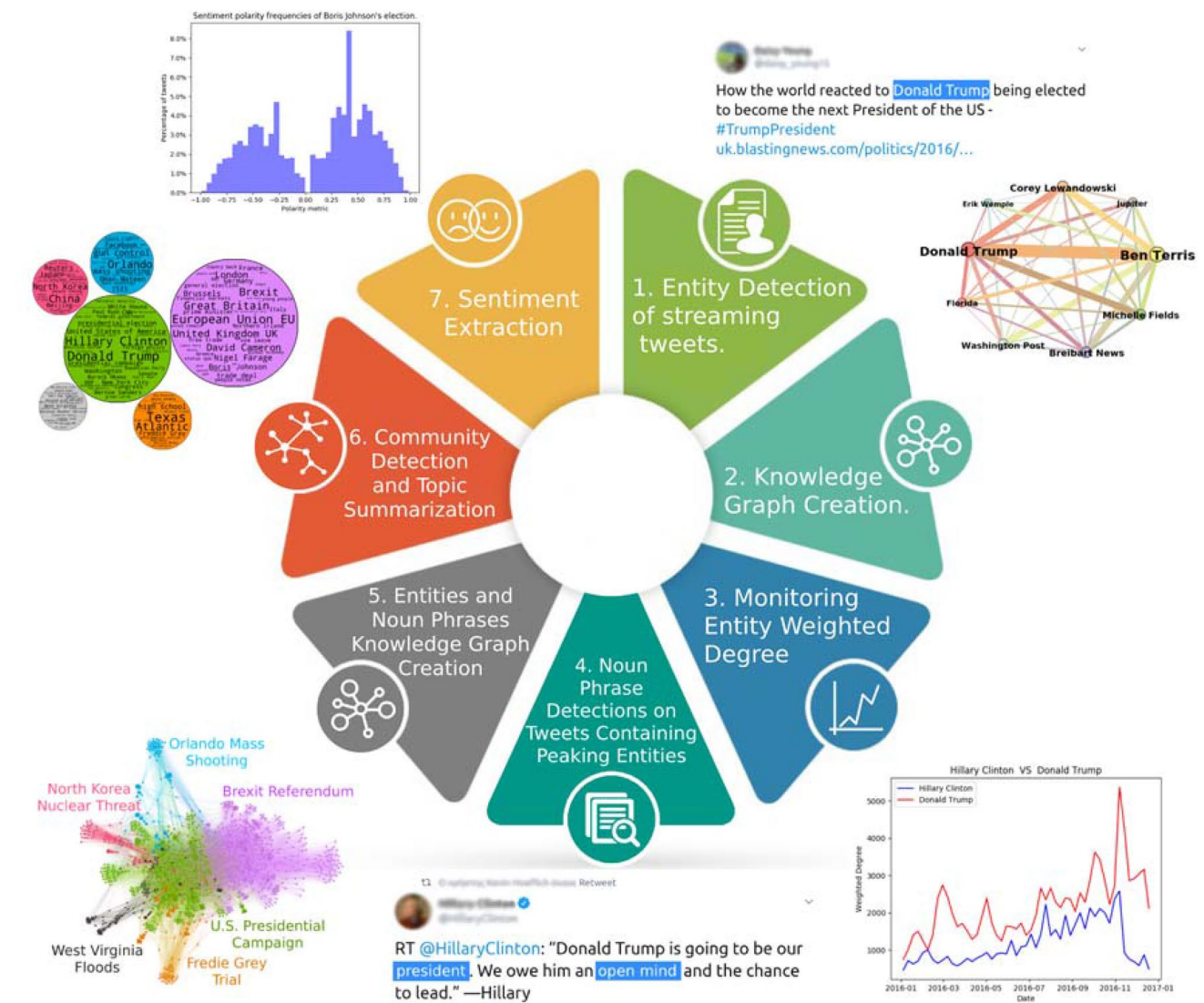


Fig. 1 Diagram of the main steps of the extended NED pipeline. The NED system receives a stream of tweets and detects peaks on the popularity of named entities, and it then gathers related to the event

documents and generates a summary. On the final step, NED extracts the sentiment of the crowd discussing the event

NED framework uses a network-based approach that links named entities to provide a holistic analysis of the ingested news stream. Named entities can be useful for detecting breaking news events since the majority of news stories tend to involve entities such as People, Locations and/or Organisations. To utilise this information, NED builds a co-occurrence graph for named entities in the document corpus, also known as knowledge graph, where each entity is a node and each edge represents how often two entities co-occur in document text. Events are identified by monitoring the weighted degree of each node using a one-day time period and identifying change points in the resulting time series. Significant peaks in an entity time series correspond to emerging (breaking) news stories. For each identified event,

all related documents are gathered and used to create an expanded knowledge graph linking named entities together with informative noun phrases. Finally, the application of community detection techniques to the expanded knowledge graph generates a set of ranked *n*-grams (named entities and noun phrases) that provides a summary description of the event. In previously published work Moutidis and Williams (2019), the NED approach is described fully and evaluated on several manually curated data sets from online news and social media.

In this paper, we expand the NED system of event detection and topic summarisation in two ways. First, we introduce a disambiguation process for single-word Person entities in tweets. This extension uses two novel features: (i) the

text of online news articles linked to URLs shared in tweets, and (ii) super documents that aggregate tweets with similar content and hashtags. Second, we evaluate the sentiment around each detected event using a set of tweets derived from that event. This addition to our method gives a simple affective measurement based on the sentiment of the crowd discussing the event.

We demonstrate the expanded NED method using three Twitter data sets from three different political elections: (i) the US Presidential election of 2012, using data from Aiello et al. (2013), in which President Barack Obama was elected; (ii) the US Presidential election of 2016, using data from Littman et al. (2016), in which President Donald Trump was elected; and (iii) the UK General Elections of 2019, won by the Conservative Party led by Prime Minister Boris Johnson. This latter data set was collected by colleagues in our research laboratory. It consists of geolocated tweets from the UK and mainland Europe gathered during the week of the General Election.

In the next section, we discuss some *Related Work* published in the literature to give context for the current study. Further sections describe the *Methodology* and *Results*, before a final *Discussion* section considers the main findings and implications of this study.

2 Related work

2.1 Topic detection and tracking

Topic detection and tracking (TDT) was first introduced in 1997 Wayne (1997); Allan et al. (1998) and incorporates a number of computer science fields, such as natural language processing, where processes like grammatical tagging, named entity recognition and term disambiguation are being used, information retrieval, where inverted indexes of document terms and clustering algorithms are utilised, social network analysis, for creating relations between linguistic entities and distinguishing network communities that constitute a topic or event and machine learning, where document features are extracted and used to classify content.

Two prevalent approaches for topic detection and tracking are exemplified by Aiello et al. (2013) and Petkos et al. (2014). The first is called ‘document pivot’, where the objective is to construct collections of documents with similar content to represent each topic and from them generate the optimal features to classify any unseen documents. The second TDT approach is called ‘feature pivot’, and its main characteristic is the detection and matching of implicit features of documents, again to create collections with similar content that represent a specific topic. Topic or event detection methodologies can also be

characterised by the mechanism used to acquire documents for analysis. Documents such as news articles or tweets can be available in real time, and related methods should be able to efficiently consume and process the incoming information Fedoryszak et al. (2019). Topic detection can be also applied retrospectively to archival collections of documents, where the method should be able to detect and characterise the important events described in the corpus, but efficiency is less important Li et al. (2005).

2.2 Sentiment analysis for Twitter data

Two main approaches for sentiment analysis of text documents are described in the literature, specifically approaches based on machine learning and approaches based on symbolic techniques Boiy et al. (2007).

Symbolic techniques use lexicons and other linguistic resources to determine the sentiment of a given text. A well-known early technique Turney (2002) used a ‘bag of words’ model (where a document is represented as a multi-set of its words, disregarding grammar and word order, but keeping multiplicity) and assigned sentiment polarity of a text based on the average polarity of the words it contains. Other symbolic approaches to sentiment analysis make use of lexical databases (e.g. WordNet Turney (2012), word-space semantic models (e.g. Pucci et al. (2009)) and knowledge bases (e.g. EmotiNet Balahur et al. (2011)) that compile dictionaries of words and associated sentiment values from large corpora. More recently, VADER Hutto and Gilbert (2014) include the word banks of established tools as well as special characters such as emoticons and cultural acronyms (e.g. ‘lol’), which makes it advantageous for social media jargon. Additionally, VADER’s model incorporates syntax and punctuation rules and is validated with human coding, making its sentence prediction 55–96% accurate.

Some research has used machine learning for classifying the sentiment of a given text, sometimes following the approach of most symbolic techniques and seeking to identify positive, negative and neutral categories, but sometimes also considering other sentiment categories such as anger, joy and sadness. Traditional machine learning methods, such as support vector machines, Naive Bayes and maximum classification entropy, can achieve high accuracy and are widely used for sentiment analysis, combined with a variety of pre-processing and feature selection approaches Pang et al. (2002); Cui et al. (2006); Davidov et al. (2010); Moh et al. (2015); Liu et al. (2013). More recently, deep learning models have been used for natural language processing tasks including sentiment analysis Dos Santos and Gatti (2014); Lan et al. (2016).

3 The network event detection (NED) system

In this work, we expand the network event detection (NED) previously introduced in Moutidis and Williams (2019). The main goal of the NED system is to detect emerging news ‘events’ from a stream of documents (articles, tweets). As a news ‘event’, we characterise a group of incoming documents related to a unique and new subject or development; this event may be considered similarly as a trending topic. The NED system detects emerging news/topics by utilising named entities detected in the document stream. The main assumption is that the occurrence of a trending topic or news event (these two terms are used interchangeably hereafter) can be detected by changes in how frequently named entities appear on the text. These named entities are Persons, Locations and Organisations.

Our system detects named entities in a stream of news documents and creates a series of networks where the nodes represent named entities and the edges represent the co-appearance of two entities in the document text. The time intervals for which networks are created can vary based on the medium being monitored (newspaper articles, Twitter) and the nature of the event. For example, when dealing with newspaper articles a network is typically created for each day, but if dealing with Twitter then the interval might be reduced to a few hours. Time series of the weighted degree of each node (entity) are created by analysing the network sequence.

The NED system consists of two main stages: event detection and event characterisation/summarisation. Emerging topics and news are spotted by detecting peaks in the named-entity degree time series. The documents associated with the time window in which an event is detected are filtered to retain those that contain ‘peaking entities’, i.e. the entities whose degree sequence revealed the peak. This decreases the amount of documents the NED system has to process, and we keep only documents that are related to the event. A second generation of networks is then created by detecting noun phrases in the filtered documents and adding them as nodes in the named entity networks to create entity-phrase networks. This permits a summary of the detected event to be created by applying community detection to the entity-phrase graph, where a community of named entities and noun phrases corresponds to a description of a news event. A more detailed description of the NED system can be found in Moutidis and Williams (2019).¹

¹ See also the pre-print available here: <https://arxiv.org/abs/2005.13751>.

3.1 Entity detection

Given a document stream, NED applies a named entity recognition (NER) technique for detecting three kinds of entities: Persons, Locations and Organisations. We used two different classifiers for this task. The first was the Stanford NER classifier Finkel et al. (2005), and it was used for entity detection on news articles, because it was trained with the CoNLL² data set which consists of Reuters newswire articles. For entity detection in tweets, we chose the classifier of Ritter et al Ritter et al. (2011). Tweets are different to news articles in that they commonly contain different linguistic features (e.g. typos, jargon and lower case letters) and are shorter in length, so the detection of entities with classifiers that are not trained in this kind of text could lack in precision. According to the authors of Ritter et al. (2011), this classifier produces significantly better results than the Stanford NER system for detecting named entities in tweets.

3.2 Knowledge graph creation

The creation of the knowledge graph utilises the named entities that the system detected. Each entity will be a node in the graph, and the co-appearance of each entity with another in the text will be an edge connecting them. The edge weights are calculated using the ‘gravity’ that each entity has in the text. To achieve that, we assign a significance value to each named entity in a given document as follows:

$$S_x(v) = \frac{tf(v, x)}{\sum_{v' \in V} tf(v', x)} \quad (1)$$

where v is the current entity (node), x is the current document (article, tweet), $tf(v, x)$ is the term frequency (the raw count of term v in text x) and V is the set of all entities in the current document. The contribution from a document d to the edge weight joining two entity nodes i, j is then given by:

$$w_d(i, j) = \begin{cases} S_d(i) + S_d(j) & \text{if } i, j \in V \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where V is the set of entities in the current document. Step 2 in Fig. 1 demonstrates an example of a created knowledge graph. The aggregation of such graphs from all documents of a given time period forms the overall knowledge graph, according to:

$$W(i, j) = \sum_{d \in D} w_d(i, j) \quad (3)$$

where D is the set of all documents.

² <http://www.conll.org>.

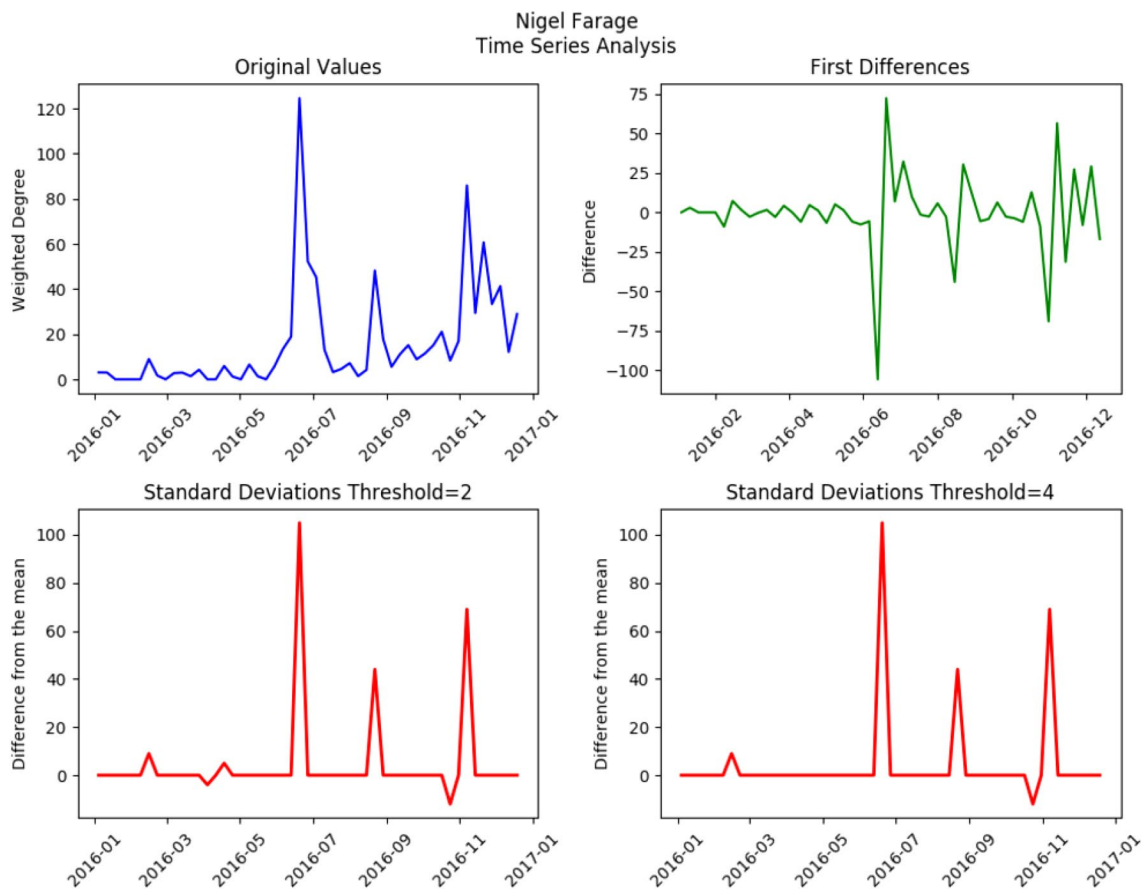


Fig. 2 Weighted degree time series for the Nigel Farage entity. The first detected peak occurs when the ‘Brexit’ referendum took place, and the second highest peak after November occurs when the US

Presidential elections took place. The two time series on the bottom show the significant peaks detected using different thresholds of standard deviations $Y = 2$ and $Y = 4$

3.3 Graph time series analysis

To detect events in a given news stream of documents, we monitor the weighted degree (i.e. the sum of the weights of all adjacent edges to the node) of all entities in the knowledge graph that we established in the previous section. Our expectation here is that a significant increase in the weighted degree of an entity indicates a news event involving that entity. We create equal duration time blocks from the incoming stream of documents and generate a sequence of knowledge graphs for each block. For every node of each graph, we create time series of their weighted degree. Our goal here is to detect significant changes on the time series caused by upcoming events.

Significant changes on the time series are detected by utilising a sliding time window. Initially, we remove trends from the time series by calculating their first differences, and then, we calculate the mean and standard deviation from a sliding window of X blocks. A ‘peaking entity’ is then identified as an entity node with weighted degree bigger than a threshold of Y standard deviations away from the rolling

mean. Figure 2 illustrates an example of the weighted degree time series analysis we described, using a window of five time blocks ($X = 5$) and a threshold distance from the mean of two standard deviations ($Y = 2$).

3.4 Summarising the detected events

The time series analysis step is followed by the characterisation/summarisation of the detected events. This is done by collecting only the documents that the peaking entity appears in the time block where the peak was detected. Depending on factors like the source of the incoming documents and the time window size, the number of detected events may vary. For that reason, we need to distinguish the occurring events to acquire useful information. Filtering out documents that do not contain peaking entities significantly prunes the amount of data our system has to process and improves the quality of the results by removing noisy input.

To identify individual events, we generate a second generation of knowledge graphs using only the filtered documents, incorporating noun phrases as well as named entities. This

kind of graphs is referred to as KeyGraphs Sayyadi et al. (2009). For the task of detecting noun phrases in the text, NED uses the ToPMine algorithm El-Kishky et al. (2014). We then apply the Louvain community detection algorithm Blondel et al. (2008) to the KeyGraph for each event period. Each detected community of nouns, noun phrases and named entities is considered as a candidate for an event. We create a bag-of-words summary of the event, sorting each bag of word using the weighted degree of each node to generate an easily interpreted synopsis of the event.

Step 5 of Fig. 1 presents an example KeyGraph where communities (events) were detected and labelled. The labels were generated by manual inspection of the top weighted entities and noun phrases as we previously described. On step 6 of Fig. 1, we present word clouds of the top entities and noun phrases for each detected community of the Key-Graph of step 5.

4 Extensions of the NED system

The software architecture and processing methods used here follow the previous implementation, described in Moutidis and Williams (2019), except in two aspects. The first alteration is an improvement to named entity disambiguation, intended to overcome the challenges of working with short texts found in tweets. The second alteration is the addition of sentiment analysis to characterise the events that are detected. Below, we describe fully the alterations to entity disambiguation and sentiment analysis. In Moutidis and Williams (2019), the NED system is evaluated for data sets consisting of both news articles and tweets, showing good event detection performance (validated against manually curated ground truth data) and effective event characterisation/summarisation. The source code of the implemented methods can be found online.³

4.1 Extension 1: entity disambiguation

Tweets contain short text strings, and the language used is often colloquial, contains slang phrases and abbreviations, as well as non-standard spellings and special characters. Furthermore, tweet text often contains typographic errors. These features present substantial challenges to named entity detection algorithms, which may recognise different variants of a single entity as distinct entities. Entity disambiguation is the process of resolving these issues.

In the original NED system Moutidis and Williams (2019), specifically in Step 1 of the pipeline shown in Fig. 1, this issue was addressed for news articles by replacing

single-word entities (usually Person entities) with their full name. To do this, each news article document was scanned for Person entities and when a single-word Person entity was detected, it was replaced by the most recent matching multi-word Person entity phrase that was found. This method works well for long-form news articles, where by convention journalists often replace a person's full name with just their surname after the first usage. Location and Organisation entities were disambiguated by replacing abbreviations. All entities were disambiguated for typographic mistakes by checking their string similarity against manually created lexicons of exceptions.

These disambiguation approaches are still useful for tweets, especially for Locations and Organisations, but the approach we followed for Person entities cannot be easily applied on tweets since their text strings are too short. To address this challenge for Person entities in tweets, we developed a new approach consisting of two phases (applied in Step 1 of Fig. 1). This is illustrated in Fig. 3. In the first phase, tweets that contain both a single-named Person entity and a URL are identified using the classifier of Ritter et al. (2011), and then, the linked content from the URL Web page is retrieved and the text content extracted using the *beautifulsoup* library Richardson (2007). All named entities are detected within the article text using the Finkel et al. (2005) classifier, which performs better and faster in larger documents such as articles. The main idea of this phase is that the linked documents will contain the entities of the tweet using both their first and last name. For that reason, our approach is to try to match single-named entities from the tweets with full-named entities on the linked document (Fig. 3). If the method detects a full-named entity that contains the single-named entity in the tweet, it replaces it on the tweet text. This approach produces the satisfactory results. Around 20% of tweets in the data sets we processed include an embedded URL (linked to news articles in most cases) from which the full name of a Person entity could be successfully identified.

In the second phase, we cluster together tweets containing Person entities based on the string similarity of their full text (using a normalised Levenshtein distance metric Yujian and Bo (2007) and the K-medoids Park and Jun (2009) clustering algorithm) and also tweets sharing one or more hashtags, creating a 'super document' representing each cluster. For each cluster, we create rankings with the frequencies of each full-named entity appearing on the super document. Next, we get every single-named entity and scan the named entity ranking of the super document. If a full-named entity contains the single-named one, we replace it. This approach further improves our disambiguation process. It is computationally efficient since it uses the named entities that were previously detected in the initial document processing stage of the NED pipeline.

³ <https://github.com/imoutidi/SNAM2020>.

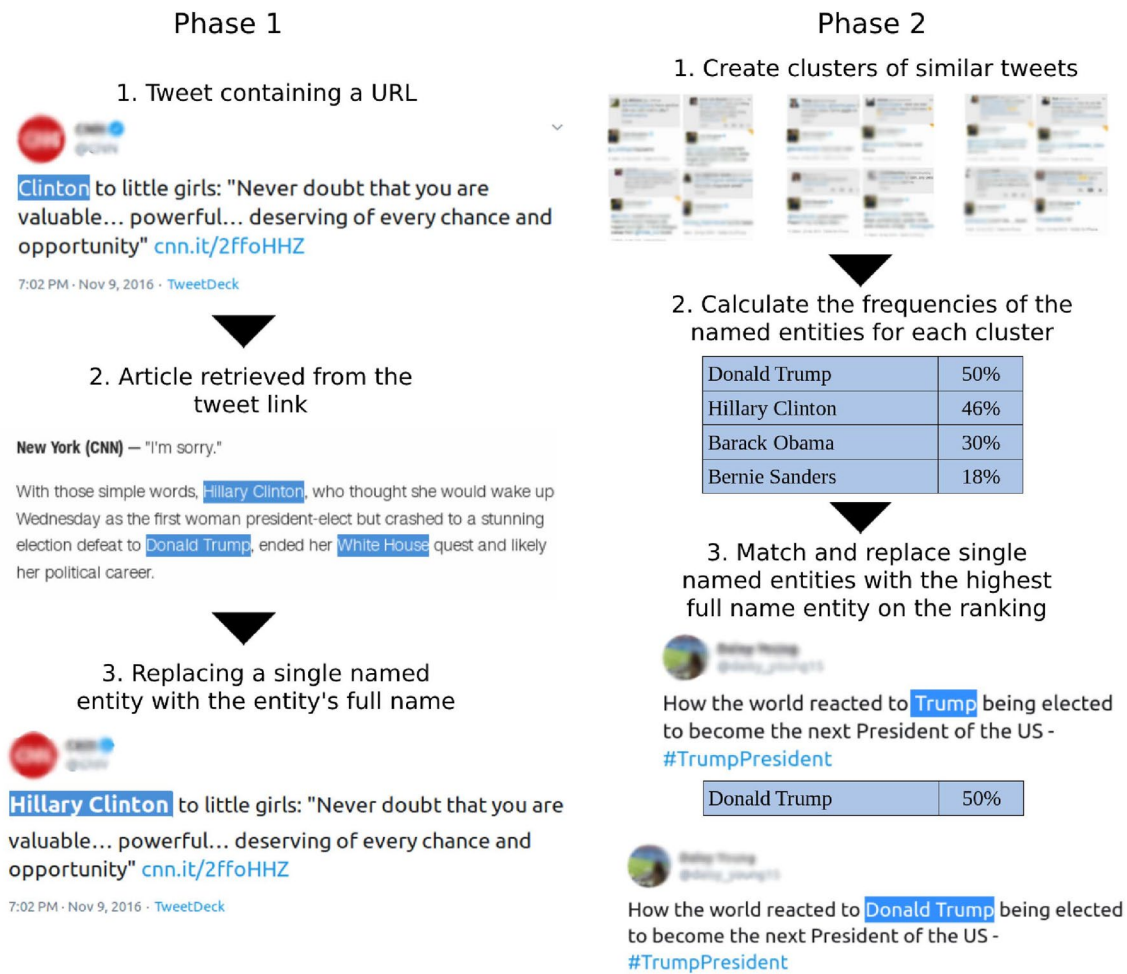


Fig. 3 Person entity disambiguation on tweets. **Phase 1 Left.** Tweet with a single-word Person entity and the linked article containing the full name of the entity. We apply named entity recognition, match the single-named entity on the tweet with the full-named entity on the article and finally replace it on the tweet. **Phase 2 Right.** All

tweets after phase 1 are being clustered based on their string similarity. Frequencies of full-named entities are calculated for each cluster, and rankings of them are created. Single-named entities are being replaced by the highest full-named entity in the frequency ranking, containing the single-named entity

4.2 Extension 2: sentiment analysis

The original NED system Moutidis and Williams (2019) produces a summary for each detected event consisting of associated named entities and noun phrases (see Step 6 in Fig. 1). The summary is a ranking of these entities based on their weighted degree on the final knowledge graph of the analysis pipeline, representing how often an entity is mentioned in the document stream and how often it co-occurs in the same text as other frequently mentioned entities.

Here, we augment this factual summarisation with a simple affective judgement of the event using sentiment analysis—effectively, and this extension seeks to determine whether the event is seen as ‘good’ or ‘bad’ by the authors contributing to the news stream. This extension (Step 7 in Fig. 1) operates by determining the sentiment in the text describing the event. To do this, all the tweets that contain

at least one of the entities or noun phrases given in the event summary are gathered in a collection. Sentiment analysis is then applied to all tweets in the collection to mine the overall sentiment distribution of the user crowd that posted tweets about the event.

For this task, the VADER model Hutto and Gilbert (2014) was used, since it addresses many of the challenges of sentiment analysis in documents coming from microblogging platforms such as Twitter, Reddit or Facebook. Key challenges arise from the small size of the text, widespread use of abbreviations, informal language and slang, as well as typographic mistakes and sarcasm. Manual comparison of the VADER model with several other algorithms on a sample of tweets showed VADER to give the best performance.

For each tweet, VADER was applied to tweet text to calculate a sentiment polarity value, using the compound metric of the model to reflect how positive, neutral or negative the

expressed sentiment is in the whole tweet. Polarity values are bounded between -1 for the most extreme negative sentiment and $+1$ for the most extreme positive sentiment. This gives a very simple unidimensional measurement of sentiment for a given sentence (or tweet). If the compound sentiment polarity score is denoted s , here we consider a tweet as positive for $s \geq 0.05$, negative for $s \leq -0.05$, or otherwise neutral ($-0.05 \leq s \leq 0.05$).

5 Evaluation data sets

To explore the effect of our improvements to the NED methodology, we applied the expanded procedure to three Twitter data sets from three different time periods corresponding to political elections:

- *US2012*. Tweets (~ 3.6 million) collected during the US Presidential Election in 2012 by Aiello et al. (2013). President Barack Obama for the Democrats was victorious over his Republican opponent Mitt Romney.
- *US2016*. Tweets (~ 5.4 millions) collected during the US Presidential Election in 2016 by Littman et al. (2016). President Donald Trump won the election for the Republicans against the Democrat candidate Hilary Clinton.
- *UK2019*. Tweets (~ 4 millions) collected within our own research group during the UK General Election in 2019. The Conservative Party led by Boris Johnson won the election and achieved a large Parliamentary majority over the Labour Party led by Jeremy Corbyn.

Within the scope of this paper, the main goal is not to detect events, since this has been demonstrated previously Moutidis and Williams (2019). Instead, we seek to test the extensions suggested above using known events that most readers will be familiar with. Being able to characterise the sentiment of a detected event adds value to the event detection process, since it informs the user about public opinion.

The US2012 and US2016 data sets were collected using election-specific keywords, so contain mostly tweets related to the elections. The UK2019 data set was extracted from a larger data set of tweets geo-located in the UK and Europe. Therefore, the UK2019 data contain tweets related to a wide range of topics. For each data set, we restrict the data to a 10-day time period spanning seven days before the day that the elections took place and two days after. The tweet IDs for all three data sets are available online.⁴

⁴ <https://drive.google.com/open?id=1a8apanZWjuoxvNXI-2O8fdznKHwIWNYM>.

Table 1 Sentiment scores for each detected election event

	Election year		
	2012 (%)	2016 (%)	2019 (%)
Positive	34	25.3	22.9
Neutral	57.8	60.6	60.4
Negative	8.2	14.1	16.7

6 Results

The extended NED method was applied to the US2012, US2016 and UK2019 data sets. The time window was set to 1-day duration, and the system successfully identified major events for all three data sets on the day following each election, when many Twitter users and news commentators were discussing the election results.

The disambiguation method was tested in a subset of the US2016 data set, containing 1700 tweets. From the 1700 tweets, 658 contained single-named entities. The first phase of the method identified 171 tweets where their url link contained a full-named entity matching with the single-word entity of the tweet it originated. It was able to successfully disambiguate 125 single-named entities from the total 171 single-named entities that were replaced ($\sim 73\%$). The remaining 46 entities were replaced with a wrong full-named entity. The second phase of the method successfully disambiguated 362 single-named entities from the total of 401 single-named entities that were replaced ($\sim 90\%$). The remaining 86 tweets containing single-named entities could not be correctly disambiguated by our method. In total, we were able to disambiguate $\sim 74\%$ of single-named entities in the evaluation data set. The disambiguation reports and evaluation data sets can be found online.⁵

The sentiment analysis component of our pipeline was applied to groups of tweets associated with each election event (see Methodology). Table 1 displays the percentage of tweets tagged as positive/neutral/negative for each election event, using the polarity ranges defined above. Sentiment is deemed negative for polarity $s \leq -0.05$ and positive for $s \geq 0.05$. This analysis shows that the majority of tweets associated with each election event were neutral ($\sim 60\%$ of all tweets). Manual inspection of these tweets suggests that the reason so many were neutral is that many Twitter users in these data sets were affiliated to, or retweeting tweets from, major news agencies like CNN, CBS, BBC, Reuters, etc. Such content tends to be moderate in its use of language and rarely expresses overt sentiment. Tweets like 'RT @ SkyNewsBreak: Exit Poll: Barack Obama and Mitt Romney

⁵ <https://drive.google.com/open?id=1SJLWSooKfXZsb3BBxWVvOjZN4za4yu8o>.

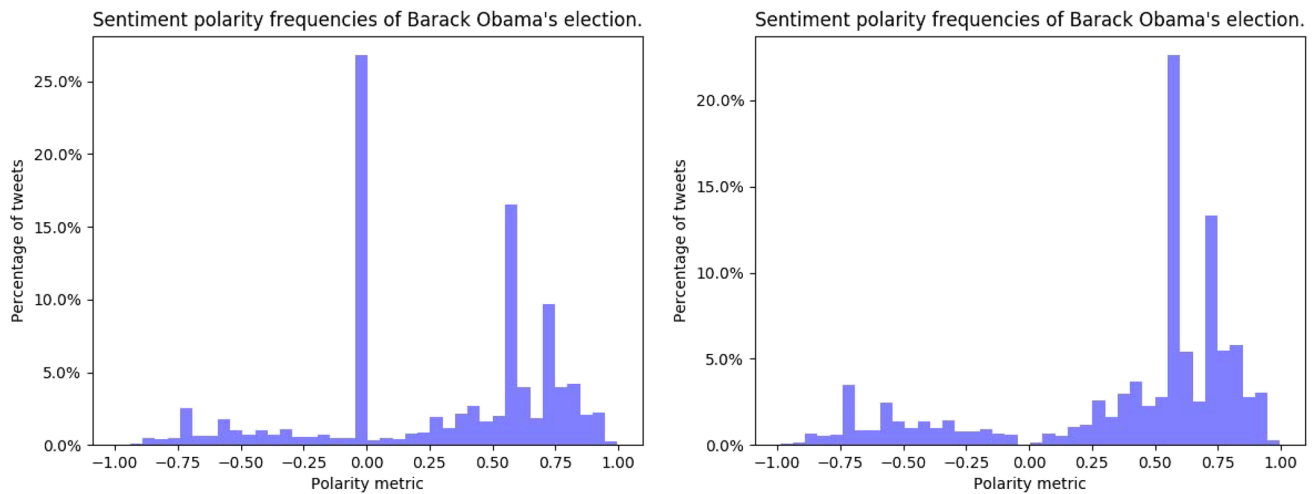


Fig. 4 Sentiment distribution of the 2012 US Presidential Elections. About 34% of tweets about the presidents election were positive against only 8.2% of negative tweets

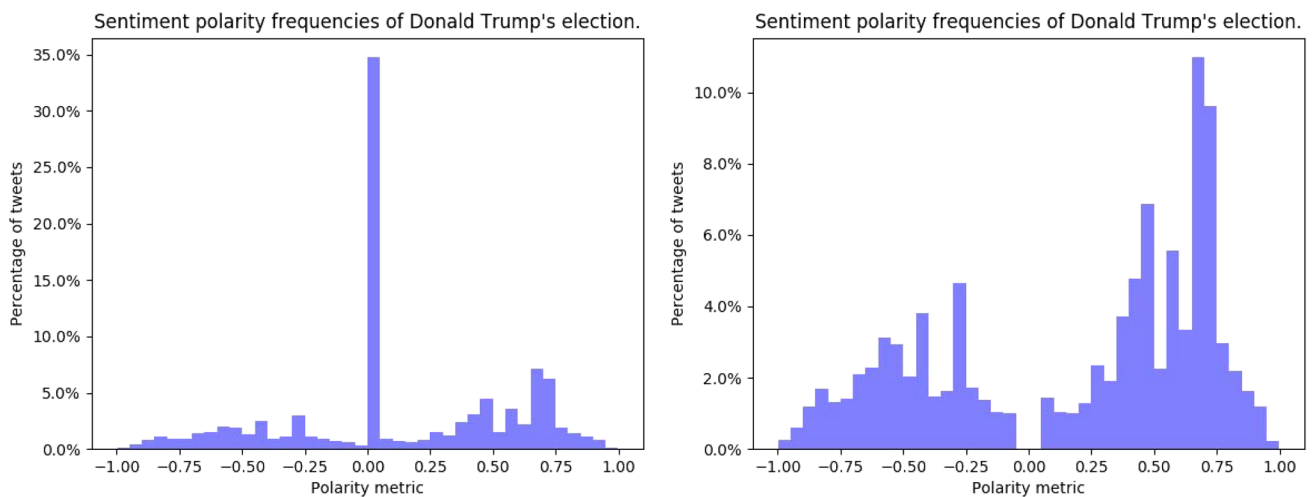


Fig. 5 Sentiment distribution of the 2016 US Presidential Elections. About 25.3% of tweets about the presidents election were positive against 14.1% of negative tweets

... tied at 49% in swing state Virginia.’ were retweeted many times, but contain no positive or negative sentiment in their language. This creates a large volume of neutral tweets. Table 1 also shows that in all cases, there are more positive than negative tweets. This may reflect user behaviour; for example, users may be more likely to tweet when they feel happy about the election outcome.

Figures 4, 5 and 6 display the distributions of sentiment polarity scores for neutral, negative and positive tweets associated with each election event. Given that the majority of tweets are neutral and often comprise tweets (and retweets) from news platforms, Figs. 4, 5 and 6 also provide the distributions of only the negative and positive sentiments for each event to permit easier inspection of the polarity scores.

Looking across all the election events, there is a bimodal distribution of sentiment polarity scores, with apparent peaks visible in both the positive and negative sides of the scale. (If plotted, the neutral tweets would create a large central spike in all three distributions.) The bimodality in sentiment distributions suggests that different user groups responded differently to the election outcomes, as might be expected in two-party political contests.

For the US2012 data set, we observe (Fig. 3) that the election of president Barack Obama was mostly received very well by the Twitter user population. The percentage of tweets with positive sentiment was 34% against only 8.2% of negative tweets. In 2012, President Obama won the election with 51.1% of the total votes and 332 electoral votes against

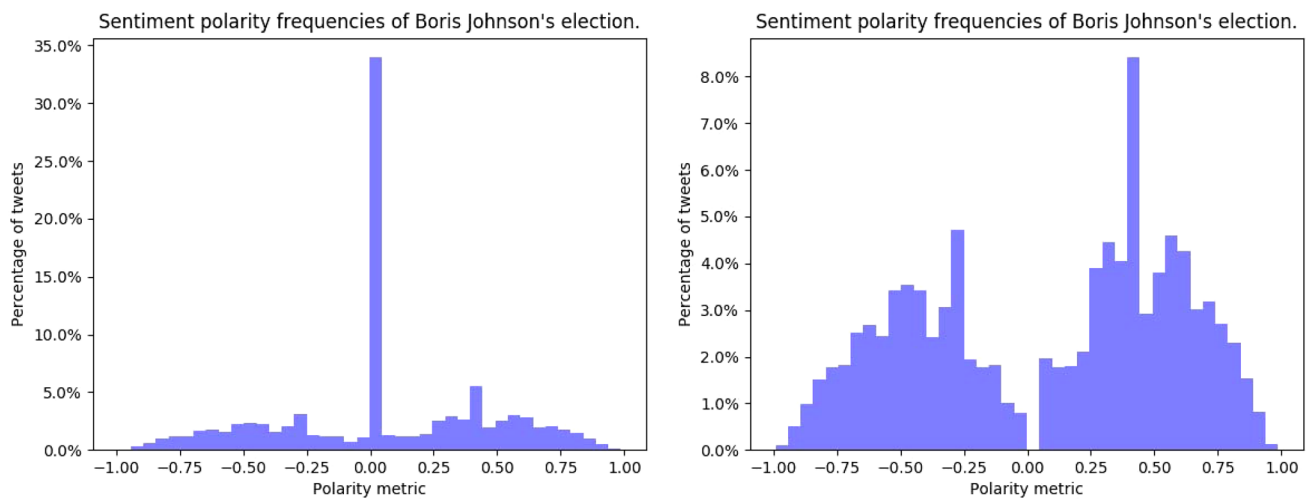


Fig. 6 Sentiment distribution of the 2019 UK General Elections. About 22.9% of tweets about the conservatives election were positive against 16.7% of negative tweets

Mitt Romney's 47.2% of the total votes and 206 electoral votes. This outcome in US2012 is perhaps more convincing and less controversial than the outcomes in US2016 and UK2019.

In the 2016 US election (Fig. 4), President Donald Trump won with 46.1% of the total votes and 304 electoral votes, against Hillary Clinton's 48.2% of total votes and 227 electoral votes. In our analysis of the US2016 data set, we infer that overall Trump's election was received positively by a big part of Twitter user population that tweeted about the election outcome, with 25.3% of the event-associated tweets being positive, but also that a substantial part of the user population (14.1%) reacted negatively.

Finally, in the UK General Election of 2019, the Conservative Party won the elections with 43.6% of the total votes, against the Labour Party which came second with 32.2% of the total votes. Tweets associated with this election event (Fig. 5) show the most balance between positive/negative sentiment, with 22.9% of tweets being positive and 16.7% of tweets being negative. It is tempting to speculate about the relationship between political polarisation during election campaigns and the balance of positive and negative sentiment expressed in social media discussions on Twitter; however, no firm conclusions can be drawn on this issue from our analysis and this topic is left for future work.

7 Discussion

This paper extends a previously developed system for news event detection and characterisation, the network event detection (NED), by adding a sentiment analysis component and improving entity disambiguation. The previous NED

system already showed good performance on event detection in dynamic news streams and characterisation in terms of identifying key entities (Persons, Locations, Organisations) and descriptive noun phrases that represent an event. The aim with these changes is to tailor the system towards social media (specifically Twitter) by improving entity resolution for short-form text documents and to improve event characterisation by determining the general perception of whether an event was positive or negative in terms of the sentiment in associated articles.

The improved entity disambiguation process primarily aims to resolve cases where a Person entity is referred to by a single name in a tweet, for example 'Hillary' or 'Clinton' referring to 'Hillary Clinton'. This causes multiplication of entities (in this case, three entities that should resolve to a single Person). Short form text such as a tweet does not contain sufficient contextual information to disambiguate these cases. Our solution takes advantage of features specific to tweets: the presence (in around 20% of tweets) of URLs linking to a longer online article relating to the tweet topic, and similarities between tweets in terms of high-information tokens such as hashtags. By utilising these features, we are able to increase contextual information around a single tweet and thereby perform effective disambiguation. This method may have utility beyond our own application area.

We demonstrated the benefit of sentiment analysis as a method of event characterisation using three case studies of political elections. The proposed methodology is able to extract the sentiment of the engaged user population about each of the detected events. For the events in question, analysis of sentiment clearly shows that while election outcomes generate a lot of positive sentiment (possibly explained by a greater tendency of users to tweet about an outcome they feel

happy about), the distribution of sentiment is bimodal. This suggests a scenario where different user groups have opposite affective responses to the election outcome, as might be expected in a major election event in a two-party democracy (and potentially exacerbated by the recent increase in political polarisation). These findings suggest potential applications for this methodology in understanding social processes more widely, seeking first to find events of public interest (using our network-based event detection method to integrate user posts from large populations) and then to characterise their impact on public sentiment and wellbeing.

Topic detection and tracking combined with sentiment analysis can be beneficial to the research of human analysts in many domains (e.g. political, financial) who can automatically detect emerging events in real time, understand what the event is about and observe the impact on public sentiment. Companies might use such methods to monitor the popularity of their products and understand consumer sentiment around their brands. Academic users for such methods might include social scientists of many kinds, including those who study politics, media effects and information diffusion.

Our methodology demonstrates one approach to the broad challenge of distilling and interpreting the rich information that is increasingly available through high-volume heterogeneous online news streams. In future work, the method might be improved by resolving sentiment down to the article level ('good' or 'bad' news reports) or to the user level ('happy' or 'sad' users or authors). Another improvement would be to identify a greater diversity of sentiment classes, beyond the simplistic positive/neutral/negative classification used here; identification of other emotions, for example joy, trust, anger or anticipation, might enable this approach to provide greater insight into public mood around unfolding news events.

Compliance with ethical standards

Competing interests The authors declare that they have no competing interests.

Author's contributions Both authors contributed equally to this manuscript. IM and HW designed the experiment. IM developed methods, wrote software and analysed data. IM and HW wrote the manuscript.

Funding The authors acknowledge funding from a commercial entity, Adarga Ltd (<https://www.adarga.ai>). The funder had no input or editorial influence over the manuscript. HW acknowledges funding from the Turing Institute. IM and HW acknowledge funding from the University of Exeter (<https://www.exeter.ac.uk>).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes

were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aiello LM, Petkos G, Martin C, Corney D, Papadopoulos S, Skraba R, Göker A, Kompatsiaris I, James A (2013) Sensing trending topics in twitter. *IEEE Trans Multimed* 15(6):1268–1282
- Allan J, Carbonell JG, Doddington G, Yamron J, Yang Y (1998) Topic detection and tracking pilot study final report
- Balahur A, Hermida JM, Montoyo A (2011) Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. *IEEE Trans Affect Comput* 3(1):88–101
- Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008(10):10008
- Boiy E, Hens P, Deschacht K, Moens M-F (2007) Automatic sentiment analysis in on-line text. In: *ELPUB*, pp 349–360
- Cui H, Mittal V, Datar M (2006) Comparative experiments on sentiment classification for online product reviews. In: *AAAI*, vol 6, p 30
- Davidov D, Tsur O, Rappoport A (2010) Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In: *Proceedings of the fourteenth conference on computational natural language learning*. Association for Computational Linguistics, pp 107–116
- Dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pp 69–78
- El-Kishky A, Song Y, Wang C, Voss C, Han J (2014) Scalable topical phrase mining from text corpora. [arXiv:1406.6312](https://arxiv.org/abs/1406.6312)
- Fedoryszak M, Frederick B, Rajaram V, Zhong C (2019) Real-time event detection on social data streams. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp 2774–2782
- Fellbaum C (2012) Wordnet. *The encyclopedia of applied linguistics*
- Finkel JR, Grenager T, Manning C (2005) Incorporating non-local information into information extraction systems by Gibbs sampling. In: *Proceedings of the 43rd annual meeting on association for computational linguistics*. ACL '05, pp 363–370. Association for Computational Linguistics, USA. <https://doi.org/10.3115/1219840.1219885>
- Hutto CJ, Gilbert E (2014) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: *Eighth international AAAI conference on weblogs and social media*
- Lan M, Zhang Z, Lu Y, Wu J (2016) Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In: *2016 international joint conference on neural networks (IJCNN)*. IEEE, pp 3172–3179
- Littman J, Wrubel L, Kerchner D (2016) United States presidential election tweet ids. <https://doi.org/10.7910/DVN/PDI71N>
- Liu S, Li F, Li F, Cheng X, Shen H (2013) Adaptive co-training svm for sentiment classification on Tweets. In: *Proceedings of the 22nd ACM international conference on information & knowledge management*, pp 2079–2088
- Li Z, Wang B, Li M, Ma W-Y (2019) A probabilistic model for retrospective news event detection. In: *Proceedings of the 28th annual*

- international ACM SIGIR conference on research and development in information retrieval, pp 106–113
- Moh M, Gajjala A, Gangireddy SCR, Moh T-S (2015) On multi-tier sentiment analysis using supervised machine learning. In: 2015 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT), vol 1. IEEE, pp 341–344
- Moutidis I, Williams HT (2019) Utilizing complex networks for event detection in heterogeneous high-volume news streams. In: International conference on complex networks and their applications. Springer, pp 659–672
- Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on empirical methods in natural language processing, vol 10. Association for Computational Linguistics, pp 79–86
- Park H-S, Jun C-H (2009) A simple and fast algorithm for k-medoids clustering. *Expert Syst Appl* 36(2):3336–3341
- Petkos G, Papadopoulos S, Aiello L, Skraba R, Kompatsiaris Y (2014) A soft frequent pattern mining approach for textual topic detection. In: Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14), pp 1–10
- Pucci D, Baroni M, Cutugno F, Lenci A (2009) Unsupervised lexical substitution with a word space model. In: Proceedings of EVALITA workshop, 11th congress of Italian association for artificial intelligence. Citeseer
- Richardson L (2007) Beautiful soup documentation. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. Accessed 7 July 2018
- Ritter A, Clark S, Mausam, Etzioni O (2011) Named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 conference on empirical methods in natural language processing. Association for Computational Linguistics, Edinburgh, Scotland, UK, pp 1524–1534. <https://www.aclweb.org/anthology/D11-1141>
- Sayyadi H, Hurst M, Maykov A (2009) Event detection and tracking in social streams. In: Third international AAAI conference on weblogs and social media
- Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, pp 417–424
- Wayne CL (1997) Topic detection and tracking (tdt). In: Workshop held at the University of Maryland On, vol 27, p 28. Citeseer
- Yujian L, Bo L (2007) A normalized levenshtein distance metric. *IEEE Trans Pattern Anal Mach Intell* 29(6):1091–1095

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.