Social Web Observatory: An entity-driven, holistic information summarization platform across sources

Leonidas Tsekouras, Georgios Petasis
Institute of Informatics and Telecommunications
N.C.S.R. Demokritos
Greece
{ltsekouras,petasis}@iit.demokritos.gr

Aris Kosmopoulos
SciFY PNPC
TEPA Lefkippos - NCSR Demokritos
27, Neapoleos, 153 41 Ag. Paraskevi, Greece
akosmo@scify.org

Abstract

The Social Web Observatory is an entity-driven, sentiment-aware, event summarization web platform, combining various methods and tools to overview trends across social media and news sources in Greek. SWO crawls, clusters and summarizes information following an entity-centric view of text streams, allowing to monitor the public sentiment towards a specific person, organization or other entity. In this paper, we overview the platform, outline the analysis pipeline, focusing on the article clustering and title extraction aspects. We then perform a user study aimed to quantify the usefulness of the system and especially the meaningfulness and coherence of discovered events in a Greek language setting, getting promising results.

1 Introduction

Entity-driven event detection and summarization is needed in real-life scenarios, such as due diligence, risk assessment, fraud detection, etc.; where the entities are usually firms or individuals.

The *Social Web Observatory* is an initiative that aims to help researchers interested in the social sciences and digital humanities study how information spreads in the news and other user-generated content, such as social media posts and comments. The overall system is composed of a back-end and a web application that provides a friendly front-end to the final users.

In this work we overview Social Web Observatory and we examine, through a human user study, a set of research questions related to its summarization performance:

- Are the event clusters created by the system meaningful, reflecting a single event?
- How well does the system avoid bringing irrelevant articles into the clusters?
- Does the system choose representative titles for the identified events?

The rest of the paper is structured as follows. In Section 2 we outline some related work and position our work. Then, in Sections 3 and 4 we describe the platform, designate the problem it is meant to face and outline the methods used in the Social Web Observatory analysis pipeline. We continue, in Section 5, by describing the experiments conducted to answer our research questions, which we then discuss in Section 6. We conclude the paper in Section 7.

2 Related Work

Our event detection is based on clustering the news articles that are found to be related to a given entity. Each cluster that results from the clustering, is considered an event. The clustering algorithm we used is agglomerative hierarchical clustering and for the similarity measure we used n-gram graphs by (Giannakopoulos and Karkaletsis, 2009), which can capture the order of n-grams in an article, taking also into account the frequency of their co-occurrence within a window. This similarity falls under the string-based measures as defined by (Gomaa and Fahmy, 2013) in their survey of text similarity measures, which means it operates on the characters of the text and does not use any external or semantic information.

Event detection can be useful for emergencies such as natural disasters, as detecting events on social media posts can give us information that may not be easily available elsewhere in order to plan the response to the emergency more effec-

Proceedings of the Multiling 2019 Workshop, co-located with the RANLP 2019 conference, pages 44–52
Varna, Bulgaria, September 6, 2019.
http://doi.org/10.26615/978-954-452-058-8_007

tively. Event detection can also help when inspecting past events. In our case we are interested in extracting events from several documents to examine what happened that is related to a specific entity. Knowing that an event happened at some specific time can help the user build a conclusion about the sentiment for the entity at that time, or why it changed. Also, using multiple documents which contain mentions of the entity for describing an event can help to further clarify its type (e.g. if an employee "left" the company to go home or was fired) and what actually happened (Hong et al., 2011).

Because of its usefulness, a lot of work has been done on event detection for textual data. For social media posts the latest works work even for real-time scenarios (Hasan et al., 2018), and as (Imran et al., 2018) note, there are additional challenges such as the latency requirements and the informal language used on such platforms.

However, we do not have to tackle these challenges, as we focus on news articles which should use more formal language and the detection is not time sensitive. There is already a delay from when the event happens to when it is reported on news websites and we can detect it on a later time to display on our application. Our focus is more in the quality of the detected events.

Neural networks have been used with success for event detection and even language-agnostic models have been developed such as (Feng et al., 2018), who tested their network on English, Spanish and Chinese.

Litvak et al. (2016) extract events from Twitter by clustering them with the EDCoW method (Weng and Lee, 2011) which they extend to improve the detection of events that unfold at the same time, a case where the wavelet analysis of EDCoW couldn't differentiate the two separate events. The user can see the top tweets, hashtags and words as a summary of the event, similar to our case, as well as a textual summary with sentences extracted from texts found in the links of the tweets that the cluster contains. There is also an interactive map with the sentiment of each country for the event.

Toda and Kataoka (2005) use document clustering based on Named Entities to tackle the problem of document retrieval for search results. They employ NER to find the important term candidates of the documents and create an index of the terms they select using two proposed criteria. Finally they categorize these terms in order to form clusters of documents. The evaluation was done on news articles, as in our case, and the results showed that users liked the categorization of the results by the Named Entities, however the authors didn't evaluate the clustering part of the system at that time.

Montalvo et al. (2015) proposed an agglomerative clustering algorithm that uses only information about the Named Entities in order to create clusters of news articles talking about the same, specific event, that can work in a bilingual setting. Other than the bilingual nature of their documents, the task is similar to our case. The existence of the same entity in the articles as well as the entity's category are both used to perform the clustering. Their results are very encouraging, and outperformed state-of-the-art algorithms.

There is also an approach by (Tsekouras et al., 2017) where the authors used just the named entities and optionally some of the more unique terms of news articles in order to cluster them into events using the k-means algorithm with a similarity matrix generated by comparing the texts with n-gram graphs. The results show that using just the named entities can make the creation of the graphs significantly faster while achieving the same or better performance than using the full text, especially on multilingual corpora.

While (Beineke et al., 2004) have defined "sentiment summarization" as selecting part of the text that best conveys the author's opinion, we consider it as creating a summary from a number of texts that talk about a specific topic while keeping the overall sentiment intact. Using the sentiment while making a summary of the documents is important, because as (Lerman et al., 2009) have found, users prefer summaries that come from sentiment-aware summarizers.

In this paper, we describe a tool that brings entity-centric, sentiment-aware, multi-document information summarization as a tool. The tool integrates a variety of intermediate analyses to fulfil its purpose, providing a unique combination of features that empower social scientists and researchers to identify and follow public trends and stances, specifically targeted to user selected entities. In the following section we overview the platform and the technologies behind it.

3 Platform Overview

The Social Web Observatory is an initiative aiming to help researchers (mainly of the social sciences and digital humanities) and journalists to study information diffusion in the social web (news and user generated content - such as comments and posts in social media networks). The Social Web Observatory listens a wide variety of news sources (more than 2000 RSS sources which post multiple news articles daily) and user generated content (such as comments in DISQUS and tweets in Twitter). Content is indexed through a search infrastructure, enabling users to retrieve context through sets of keywords, for further analysis. Content retrieved through keyword search is analysed along various dimensions to extract indicators such as trends, coverage, events, sentiment, stance, etc. Both context and indicators are visualised through predefined dashboards and other analytics tools, to provide information and insights on the various issues defined by keyword searches.

The Social Web Observatory web application allows users to create an account and define entities with public or private access, for which dashboards are created. Each entity is comprised of a title, a type (which may allow the user to add additional fields, such as first, middle and last name) and some optional fields such as their social media information and URLs for the entity's web, Wikipedia and Wikidata pages. The user can also specify keywords to include in the search for the entity, such as alternative names or nicknames that people use to refer to the entity and keywords to exclude from the search, which can be useful if for example a last name of an entity is also a word in that language. An entity being "public" means that all users of the application can view the dashboard for that entity (but only the owner can edit it), while "private" means that only the creator of the entity is aware of its existence and can see its dashboard or edit it. A screenshot of the entity creation screen of the application can be seen in Figure 1.

The dashboard of an entity tries to show an overview of what is being said related to the entity on the web over a given date range, which the user can change. It contains information such as how many articles, comments and tweets related to that entity have been collected over the selected time period and how many unique domains had articles and comments about the entity. Then there

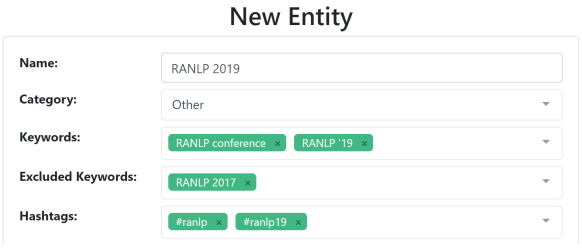

Figure 1: Part of the entity creation screen of the web application.

are tabs for more specific information about the news articles, comments and tweets about the entity, which contain a number of charts. The "sentiment over time" chart shows how the number of positive, neutral and negative documents (whose type depends on the selected tab) changes over the selected time period. For news articles, we also display the automatically detected events on the chart. The user can click an event to reveal more information about it. The user can also click a point on the chart to reveal the titles of the documents that correspond to that time point and view them at their source web page. Each of the articles, comments and tweets tabs also contain a graph that shows how many of the total collected items in each case were found to contain the entity over the same time period. This shows how much of the web is concerned with that entity at a given time.

The back-end gathers news articles from a variety of RSS sources, crawls some of the news websites to gather comments for their articles or through DISQUS, and receives tweets from Twitter. These news articles, comments and tweets are all analyzed to identify any entities that they contain, obtain their overall sentiment as well as the sentiment for each of the mentioned entities. Finally the news articles are clustered in order to form events. Since we perform named entity recognition (NER) on the articles from which the events are formed, each event can be linked to the entities that are mentioned in the articles that it contains.

4 Proposed System

The research problem which the SWO platform faces is the following. Given

- a set of text streams $\mathbb{S}$,
- a set of surface representations (i.e. alterna-

tive wordings) of an entity $\mathbb{E}$,

- a time span $\mathbb{T}$,

we are called to provide a list $\mathbb{L}$ of events, published within the time span $\mathbb{T}$, referring to the entity $\mathbb{E}$ and annotated by the sentiment expressed therein. The events should ideally be identified by a representative title and should be mapped to (i.e. supported/explained by) a number of texts from the input text streams $\mathbb{S}$. To face this problem, the Social Web Observatory project combines a number of approaches into an analysis pipeline, as described below.

The pipeline for the creation of events from the news articles is supported by the Elasticsearch (Gormley and Tong, 2015) database and begins with the news gathering. This is done through crawling a custom list of over 2000 RSS feeds one by one, receiving the available news articles from each feed and adding the ones that we don't already have to the Elasticsearch index where we keep all the articles. This process is run every 20 minutes on our server.

Periodically, we run the next step of the pipeline, entity detection and aspect-based and document-level sentiment analysis (Petasis et al., 2014; Papachristopoulos et al., 2018). This begins by taking as input the latest raw news articles/comments/tweets from the gathering step, processing and saving them in another index where we keep the processed news articles. The processing starts by detecting any entities that are in the text. For this purpose, the keywords provided by users are primarily used (for direct matching), in cooperation with an automated NER system (OpinionBuster (Petasis et al., 2014)) for some predefined types of entities, such as persons. News articles that contain entity mentions are kept for further processing. Then, the overall sentiment of each textual artifact is found as well as the sentiment for each of the entity mentions that were found in the text. For sentiment analysis, OpinionBuster (Petasis et al., 2014), a state-of-the-art system for the Greek language is being used. OpinionBuster employs a rule-based approach for performing polarity detection, based on compositional polarity classification (Klenner et al., 2009). It analyses the input texts with the aid of a polarity lexicon that specifies the prior polarity of words, which contains more than 360,000 unique word forms (Greek is an inflectional language) and more than 35,000 phrases. As a second step, the latest versions of Ellogon's (Petasis et al., 2002) dependency parser and chunker are used to determine dependencies and phrases that are the basis for a compositional treatment of phrase-level polarity assignment. Once polarity has been detected, it is distributed over the involved entity mentions with the help of dependencies originating from verbs, in order to distinguish whether the entity mentions receive or generate the polarity detected in the phrases. In case, however, a verb is encountered that cannot be handled by a rule then a simple heuristic is applied, which assigns the detected polarity to all entity mentions within the phrase. At the end of the sentiment analysis step, we have articles, comments, and tweets with the entities that they mention, the overall sentiment and the sentiment for each of the entities (calculated by summing the sentiment for each of the entity's occurences).

The last step is clustering the news articles into events. The input for this step is the processed articles, and the output the clusters, each of which represents an event. The events are saved in another Elasticsearch index that is read by the web application in order to display the events to the user. We assume that most news events should happen at daytime, so we run the clustering on the articles of each day individually. This means that if an event starts in one day and ends the next, we might miss or cluster it as two separate events. The clustering service starts the clustering for each day when that day has passed and all articles that were gathered within that day are processed by the previous step.

The clustering uses n-gram graphs (Giannakopoulos and Karkaletsis, 2009) to create a representation of each news article, which are then compared with each other in order to calculate the similarity matrix between all the texts. The news items are clustered using a modified version of the NewSum (Giannakopoulos et al., 2014) clustering algorithm. The original NewSum clustering represented each text with an n-gram graph and grouped together documents that surpassed a heuristically-defined threshold of similarity (specifically Normalized Value Similarity, which takes into account the overlap between graph edges and their relative weights (Giannakopoulos and Karkaletsis, 2009)). Thus, if a the similarity sim of a text a to a text b exceeds the threshold T, then: $\{a, b\} \in C$, where C is a cluster (i.e. set of texts). The caveat was that in several cases a was marginally, but sufficiently

similar to b, which in turn was marginally, but sufficiently similar to a text c. This meant that a, b, c would belong to the same cluster C, even though a and c had almost nothing in common. Essentially, the algorithm did not enforce coherence across all pairs within the same cluster.

In the SWO version of the algorithm an agglomerative hierarchical clustering algorithm which ascertains a minimum coherence (i.e. variation of similarity) across all pairs within a cluster was employed to produce clusters of articles. Essentially, the hierarchical clustering only adds articles to a cluster, if they have sufficient similarity to all cluster articles. This causes smaller, more coherent clusters, and prefers precision (keeping clusters clean) over recall (bringing in the maximum number of related news).

The system also extracts a title selected from the articles contained in the cluster, following a centroid-based approach: after representing all the article titles as a bag-of-words in a vector space, the system chooses the title which is closest to the centroid of all the article titles in this space.

Thus, through the clustering process, the clusters have a title and the IDs of the news articles which they contain. After the clustering runs, we need to find out which entities are related to each cluster (event) so we can later filter them by their entities. This will allow us to show only the events that are relevant to an entity in its dashboard page. To do that, we get the unique article IDs from all the clusters that were produced, retrieve them from the processed news articles index, and for each cluster we gather all the entities from all its articles and save them together with the other information about the cluster to the events Elasticsearch index.

The events then can finally be viewed on the web application in the "sentiment over time" chart of an entity's dashboard, as shown in Figure 2. Each colored plot band on the chart represents an event, starting and ending at the first and last publication times of its articles respectively. The chart shows the 50 largest events in the selected time period measured by the number of articles they contain (cluster size). By clicking on an event, the user is shown its title, start and ending times, as well as the sentiment distribution of the event's articles (i.e. how many positive, neutral and negative articles are in the event). The navigator control at the bottom of the chart helps the user click events with very small timespans by allowing them to zoom in.

5 Experiments

In order to evaluate if the events we create are coherent and if they can be labeled consistently by different humans, we ran a user study with three annotators. The annotators (Greek natives) were shown the title and articles of each event in Greek and were asked three questions each time:

- Do the articles of the cluster appear to represent a single event? (Yes/No)
- How many articles do they feel are irrelevant to others? (Number between 0 and the total articles of the cluster)
- Does the cluster (event) title reflect the event well? (Badly/Barely Acceptably/Well enough)

The data we used were the 30 events that contained the most news articles in the time period between July 1 and July 14 of 2019. This data, containing the event titles, date ranges and their articles with publication date, sentiment analysis/NER results and text content is available upon request.

With the answers of the annotators, we can then run statistical tests in order to see the inter-annotator agreement, as well as how the event clustering performs.

For the inter-annotator agreement we ran two different tests. First, we ran paired t-tests between all annotator pairs for the number of articles that they found irrelevant in the events, in order to see if there is a statistically significant difference between their answers. We also ran a chi-squared test with the two categorical variables being the annotator ID and their answers on whether they felt that the cluster's articles represented a single event. This test will show us whether there is a dependence of the result (answer) and the annotator, or whether the annotators seem to provide similar answers.

To see if the clusters are coherent, we studied how many irrelevant articles were found in each cluster by the annotators as a percentage of the cluster size and also the cluster size distribution, to support the cluster coherence result.

6 Results

In this section we will present the results of the described experiments for each set of experiments,

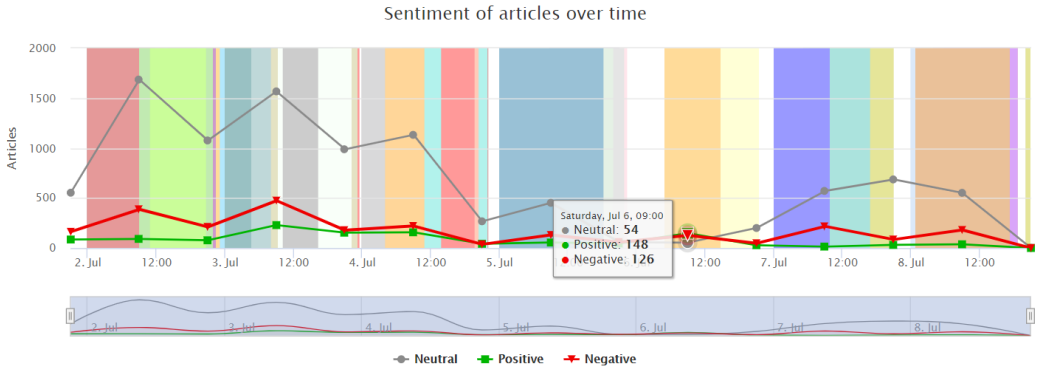

Figure 2: The "sentiment over time" chart for articles, with the colored bands representing events.

indicating how they answer our original research questions posed in Section 1.

Essentially, we examined the event cluster coherence (first two questions) and the title assignment quality (third question). Below, we describe how we ascertained that the study was meaningful and the results we got.

6.1 Inter-annotator Agreement

Our first challenge is to show that annotators can consistently judge the system. We first performed a chi-squared test to show if the annotators agree on whether each cluster represents a single event, the results show that there is no statistically significant dependence between the annotator and the resulting answer for the question (p-value $= 0.81$). Therefore, we can say that the annotator's answers are independent from the individual annotators, that is, the events seem to get the same answer regardless of who is the annotator.

We, also, performed a set of paired t-tests between the annotators to show whether the distributions of errors (irrelevant articles) identified by each annotator on each event were different. The tests showed that there is no statistically significant difference between any pair of annotators (all p-values are $> 10\%$, see Table 1). This means that the annotators seem to agree on how many articles are irrelevant in each cluster, which indicates a consistent evaluation process.

Given the above findings, we can consider the evaluation task meaningful enough to provide useful feedback.

6.2 Clustering Coherence

To analyze the coherence of the clusters, we made two plots. The first one (Figure 3) shows the cluster coherence according to our annotators, meaning how frequently we find clusters with a certain percent of irrelevant articles, according to the annotators' judgement. We see that in over 90% of the clusters the percentage of irrelevant articles that are contained in the cluster was perceived to be less than 10%. There is a very small percentage of clusters (around 2%) where the irrelevant articles make up 10-20% of the cluster. Around 5% of the clusters contain around 30-40% irrelevant articles. There are some more clusters that have around 60-70% irrelevant articles in them, but that is also a very small amount (around 2%). This shows that, overall, most clusters have a very low amount of irrelevant articles in them. At this point we should note that high percentages of irrelevant articles within clusters could also be attributed to small clusters, where a single error could amount to a big percentage of error (our error analysis indicated that this was the case).

Annotator Pair	p-value
A & B	0.1033
B & C	0.3256
A & C	0.4235

Table 1: p-values of paired t-tests between the three annotators.

We next studied the cluster size distribution to better understand if the clusters were also useful (i.e. non-trivial, having only 1 article). For each cluster size (article count contained), we see how many clusters of that size exist in our evaluated data. Looking at the cluster size statistical summary (quartiles) in Table 2, we see that the minimum number of articles found in any cluster is 3. Combining this with Figure 4, we observe that almost half of the clusters are small, but non-trivial,

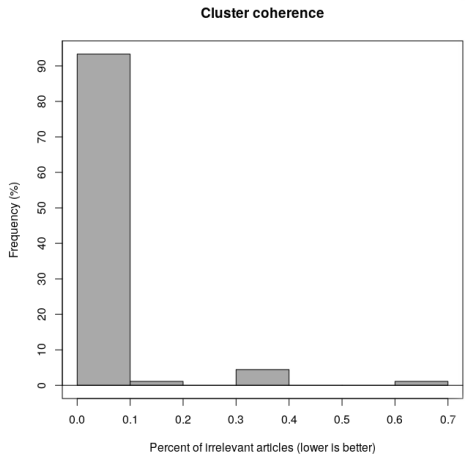

Figure 3: Clustering coherence according to the annotators.

Min	1st Qu.	Median	3rd Qu.	Max
3	3	5	7,75	26

Table 2: Basic statistical summary of cluster sizes.

meaning they contain 3-5 articles. The other half has over 5 articles (the median is 5 articles), in some cases even containing more than 20. Therefore, we can draw the conclusion that the clusters seem to be coherent, meaningful and useful.

We have to note that this evaluation takes into account only the precision of the clustering, as we cannot draw any conclusions about the recall. However, previous works (Giannakopoulos et al., 2014) have suggested that having better precision in such a task gives more perceived value for the user than recall. That is, users prefer small, clean clusters than larger clusters which may contain more of the relevant articles but also more off-topic articles.

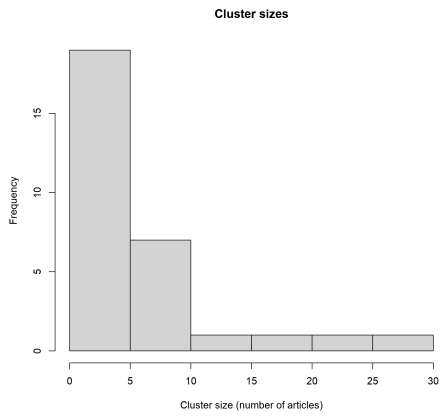

Figure 4: Cluster size distribution.

We also measured the average perceived appropriateness of a title for a given cluster, by assigning the value 0 to ”badly”, 1 to ”barely acceptably” and 2 to ”well enough”. In our data, in 23 of the 30 events (76% of the cases) the quality was at least 1 on average. In 50% of the overall events the title was considered good enough. Thus, the users seem to be able to understand what events are about from their title.

In the final section of this work we summarize what we did and suggest future steps.

7 Conclusion

In this work, we presented Social Web Observatory, an initiative that aims to show how information is diffused and spread in the social web, via a web application and a back-end system which analyzes the gathered data. Part of this system is using event detection to show events to the user, in order to help them explain why the sentiment about an entity may have changed at a given time. The event detection is run on the news articles of each day, which are analyzed for sentiment and entity recognition. On the user study that we performed, the annotators seemed to agree that the clusters contained very little irrelevant articles, which means the overall pipeline is suitable for our use case. Furthermore, we saw that the title extracted and assigned to each event is in more than 75% of the cases at least acceptable.

As future work, we want to improve the scalability of the overall pipeline to allow it to run on a larger amount of articles, as we continue to increase the number of RSS feeds that we monitor over time. Because we run the event detection periodically (once per day), in this work we were not concerned with its speed, so there is room for improvement in that area. For example we could employ blocking techniques as they have shown to significantly improve the scalability of document clustering in (Pittaras et al., 2018) without hurting the performance too much.

Acknowledgments

We acknowledge support of this work by the project “APOLLONIS: Greek Infrastructure for Digital Arts, Humanities and Language Research and Innovation” (MIS 5002738) which is implemented under the Action “Reinforcement of the Research and Innovation Infrastructure”, funded

by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. 2004. Exploring sentiment summarization. In *Proceedings of the AAAI spring symposium on exploring attitude and affect in text: theories and applications*. The AAAI Press Palo Alto, CA, volume 39.

Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. A language-independent neural network for event detection. *Science China Information Sciences* 61(9). https://doi.org/10.1007/s11432-017-9359-x.

George Giannakopoulos and Vangelis Karkaletsis. 2009. N-gram graphs: Representing documents and document sets in summary system evaluation. In *Proceedings of Text Analysis Conference TAC2009 (To appear)*.

George Giannakopoulos, George Kiomourtzis, and Vangelis Karkaletsis. 2014. Newsum:"n-gram graph"-based summarization in the real world. In *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*. IGI Global, pages 205–230.

Wael H Gomaa and Aly A Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications* 68(13):13–18.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. 2018. A survey on real-time event detection from the Twitter data stream. *Journal of Information Science* 44(4):443–463. https://doi.org/10.1177/0165551517698564.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 1127–1136.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2018. Processing Social Media Messages in Mass Emergency: Survey Summary. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*. ACM Press, Lyon, France, pages 507–511. https://doi.org/10.1145/3184558.3186242.

Manfred Klenner, Stefanos Petrakis, and Angela Fahrni. 2009. Robust compositional polarity classification. In *Proceedings of the International Conference RANLP-2009*. Association for Computational Linguistics, Borovets, Bulgaria, pages 180–184. http://www.aclweb.org/anthology/R09-1034.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09*. Association for Computational Linguistics, Athens, Greece, pages 514–522. https://doi.org/10.3115/1609067.1609124.

Marina Litvak, Natalia Vanetik, Efi Levi, and Michael Roistacher. 2016. Whats up on twitter? catch up with twist! In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. pages 213–217.

Soto Montalvo, Raquel Martnez, Vctor Fresno, and Agustn Delgado. 2015. Exploiting named entities for bilingual news clustering: Exploiting Named Entities for Bilingual News Clustering. *Journal of the Association for Information Science and Technology* 66(2):363–376. https://doi.org/10.1002/asi.23175.

Leonidas Papachristopoulos, Pantelis Ampatzoglou, Ioanna Seferli, Andriani Zafeiropoulou, and Georgios Petasis. 2018. Introducing sentiment analysis for the evaluation of library's services effectiveness. In *Proceedings of the 10th Qualitative and Quantitative Methods in Libraries International Conference (QQML2018)*. Chania, Greece.

Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Ion Androutsopoulos, and Constantine D. Spyropoulos. 2002. Ellogon: A New Text Engineering Platform. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*. European Language Resources Association, Las Palmas, Canary Islands, Spain, pages 72–78. http://www.ellogon.org/petasis/bibliography/LREC2002/LREC2002$petasis.pdf$.

Georgios Petasis, Dimitris Spiliotopoulos, Nikos Tsirakis, and Panayotis Tsantilas. 2014. Sentiment analysis for reputation management: Mining the greek web. In Aristidis Likas, Konstantinos Blekas, and Dimitris Kalles, editors, *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings*. Springer, volume 8445 of *Lecture Notes in Computer Science*, pages 327–340.

Nikiforos Pittaras, George Giannakopoulos, Leonidas Tsekouras, and Iraklis Varlamis. 2018. Document clustering as a record linkage problem. In *Proceedings of the ACM Symposium on Document Engineering 2018*. ACM, page 39.

Hiroyuki Toda and Ryoji Kataoka. 2005. A search result clustering method using informatively named entities. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, pages 81–86.

Leonidas Tsekouras, Iraklis Varlamis, and George Giannakopoulos. 2017. A graph-based text similarity measure that employs named entity information. In *RANLP*. pages 765–771.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in twitter. In *Fifth international AAAI conference on weblogs and social media*.