Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2000–2008
Marseille, 11–16 May 2020

Social Web Observatory: A Platform and Method for Gathering Knowledge on Entities from Different Textual Sources

Leonidas Tsekouras*, Georgios Petasis*, George Giannakopoulos*†, Aris Kosmopoulos†
*Institute of Informatics and Telecommunications, †SciFY PNPC – TEPA Lefkippos
National Centre for Scientific Research (N.C.S.R.) "Demokritos",
P.O. BOX 60228, Aghia Paraskevi, GR-153 10, Athens, Greece.
{ltsekouras, petasis, ggianna}@iit.demokritos.gr, akosmo@scify.org

Abstract

Within this work we describe a framework for the collection and summarization of information from the Web in an entity-driven manner. The framework consists of a set of appropriate workflows and the Social Web Observatory platform, which implements those workflows, supporting them through a language analysis pipeline. The pipeline includes text collection/crawling, identification of different entities, clustering of texts into events related to entities, entity-centric sentiment analysis, but also text analytics and visualization functionalities. The latter allow the user to take advantage of the gathered information as actionable knowledge: to understand the dynamics of the public opinion for a given entity over time and across real-world events. We describe the platform and the analysis functionality and evaluate the performance of the system, by allowing human users to score how the system fares in its intended purpose of summarizing entity-centered information from different sources in the Web.

Keywords: Knowledge Discovery/Representation, Opinion Mining / Sentiment Analysis, Tools, Systems, Applications

1. Introduction

The *Social Web Observatory*[1] 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. The platform allows users to define their own entities via a simple user interface and can then show a summary of the information that has been gathered about the entity.

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.

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..

[1] https://socialwebobservatory.iit.demokritos.gr/

2. Related Work

The proposed event detection is based on clustering of news articles which are related to a given entity. In our approach each cluster is considered an event. We combine agglomerative hierarchical clustering with n-gram graphs by (Giannakopoulos and Karkaletsis, 2009) as a similarity measure, which capture the order of n-grams in an article and take 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 used during emergencies, such as natural disasters, in order to respond more effectively. Detecting events on social media posts provides such information, which can not be easily available elsewhere. In our case we wish to examine what happened by extracting events from several documents, which are related to a specific entity. By knowing that an event happened at some specific time the user is able to build a conclusion about the sentiment for the entity at that time, or why it changed. Furthermore, by using multiple documents mentioning the entity, in order to describe an event, helps to 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).

A lot of work has been done on event detection for textual data due to its usefulness. For social media posts the latest works handle even real-time scenarios (Hasan et al., 2018) with the additional challenges that these come with, such as the latency requirements and informal language used on such platforms (Imran et al., 2018).

However, by focusing on news articles, we do not have to tackle these challenges, since a more formal language is used and the event detection is not time sensitive. Given that there is already a delay between an event and its re-

porting on news websites, we do focus on detecting it as soon as possible, but on 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). They extend EDCoW to improve the detection of events that unfold at the same time, a case where its wavelet analysis could not differentiate the two separate events before. 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 extracted from texts found in links of the cluster's tweets. 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 Named Entity Recognition 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 at the time.

In another approach by (Tsekouras et al., 2017) the authors use just the named entities and optionally some of the more unique terms of news articles in order to cluster them into events. The clustering is done with the k-means algorithm and a similarity matrix generated by comparing the texts with n-gram graphs. The results show that using just the named entities makes 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, which builds upon the work of (Tsekouras et al., 2019), we provide more details about the Social Web Observatory platform and focus more on the various available functionalities. We describe a usage scenario from start to finish showing how a user can take advantage of the platform and the analytics it provides to view and understand the opinion for an entity across the Web. Finally, we extend the experimental evaluation of the previous work with a second dataset and ask our annotators to provide more detailed data in order to better understand the quality of the platform's event detection.

(Leban et al., 2014) have created a similar system that gathers news articles from the web and identifies events through clustering. An online clustering algorithm is used, combined with a vector representation of the texts. Furthermore, in their representation more focus is given on entities detected by a named entity recognizer. One difference in SWO is that we use n-gram graphs by (Giannakopoulos and Karkaletsis, 2009) as the text representation for event clustering. Another one is that we work with Greek texts, while they use articles in four other languages. SWO's approach in general is more entity-centric. We start by defining entities of interest and use then in order to filter the articles. Another difference is that we gather documents from more sources: RSS feeds with news articles, comments and tweets. Sentiment analysis also plays an important role, as all documents found containing the entity are analyzed for sentiment, before displaying the results in our web application. In the following section we overview the SWO 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 to 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, using a search infrastructure, enabling the users to retrieve context through sets of keywords. The retrieved context is analysed along various dimensions and several indicators are extracted 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 a user to create an account and define publicly or privately accessible entities. Each entity is comprised of a title, a type (which may allow the user to add additional fields, such as the first, middle and last name of a Person) and some optional fields such as their social media information and URLs for the entity's web, Wikipedia and Wikidata pages. There are also fields allowing the addition of keywords to be included or excluded during an entity search. Inclusion of keywords can be used to provide alternative names or nicknames that people use to refer to the entity. Exclusion of keywords 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 are allowed to view the dashboard for that entity (but only the owner can edit it), while "private" means that only the creator of

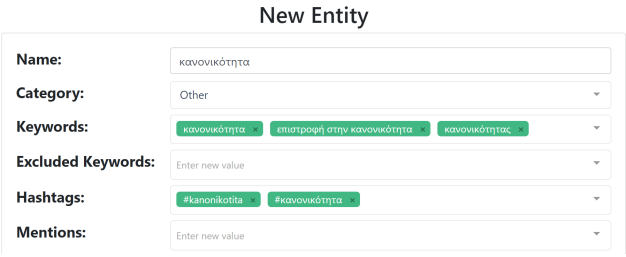

Figure 1: Part of the entity creation screen of the web application.

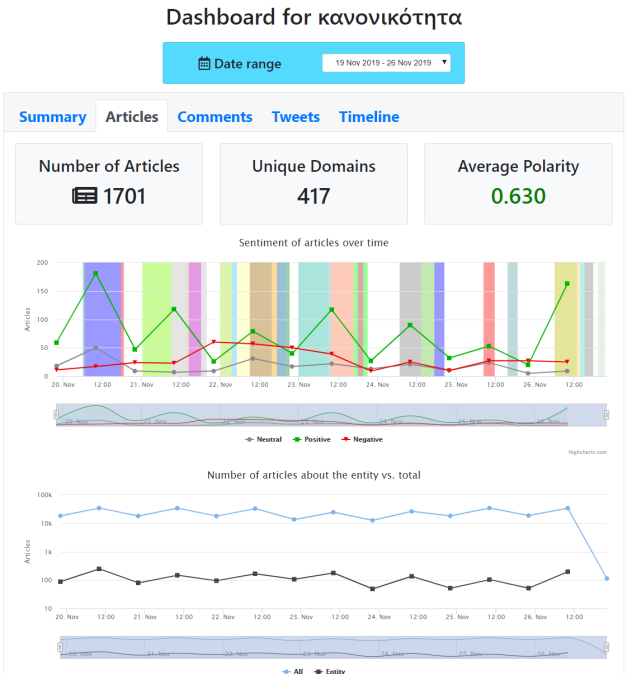

Figure 2: Dashboard of an entity.

the entity is aware of its existence and able to see its dashboard or edit it. Figure 1 shows a screenshot of the entity creation screen of the application.

The dashboard of an entity, shown in Figure 2, tries to show an overview of what is being said related to the entity on the web over a given date range. It contains information on how many articles, comments and tweets related to that entity have been collected over the selected time period. It also displays the number of unique domains that have articles and comments about the entity. Furthermore, a specialized tab per source type (news articles, comments and tweets) is provided, containing a number of charts.

The "sentiment over time" chart (Figure 3) shows the fluctuation in the number of positive, neutral and negative documents over the selected time period. For the news articles we also display the automatically detected events on the chart. By clicking an event more information is revealed about it. Clicking a point on the chart shows a panel with the titles of the documents that correspond to that time point. A link is also provided to the document's source web page. Each tab contains a graph showing how many items were found containing the entity, over the selected time period. Such a graph shows how much of the web is concerned with the entity, at the given time.

Finally, we have the Timeline tab that shows all the events related to the entity for the selected date range on an interactive timeline, outside of the "sentiment over time" chart. The advantage of using such a visualization is that having the event titles visible on the timeline itself, provides a quick overview of the events simply by scrolling.

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 on the articles contained in the events, each event is linked to the entities found in its articles.

In the "Sources Overview" page of the application we have an overview of how many items have been crawled by the back-end, with a layout similar to that of Dashboards. There are charts about how many articles each crawler found in the chosen date range, the number of articles and comments per domain, as well as a list of all the data sources sorted by the amount of items found in each. In the "Articles", "Comments" and "Tweets" tabs the sentiment chart from Dashboards has been replaced with a chart that shows the number of crawled items per source type over time.

3.1. Potential Users of the Platform

SWO finds practical application in professionals of various industries, such as journalism and advertisement. In the field of journalism, a professional using the platform can easily survey the popularity of an entity in the social web.

By identifying which channels provide the most information regarding an entity, a journalist might deduce several things. For example, the journalist could estimate which ages are more interested in the entity, by taking into account the age groups that usually visit these web sources. By comparing the metrics regarding an entity before and after a specific event, such as a publication or defamation, someone could investigate how that event affected specific organisations or individuals.

Furthermore, a researcher can examine if a topic is a trend during a particular point in time (e.g. deadly epidemics) and choose the right time to publish relevant content and surveys in order to attract a larger audience. Essentially, the platform helps saving time and resources by gathering the requested information from thousands of web sources and displaying it in a structured way.

The platform could also be used to study the effect of governmental, social, legislative and other decisions on the citizens. It congregates a significant amount of reactions by internet users, that can be studied to assess the general opinion. During election periods for example, the platform can help in estimating the popularity of a specific politician or political party.

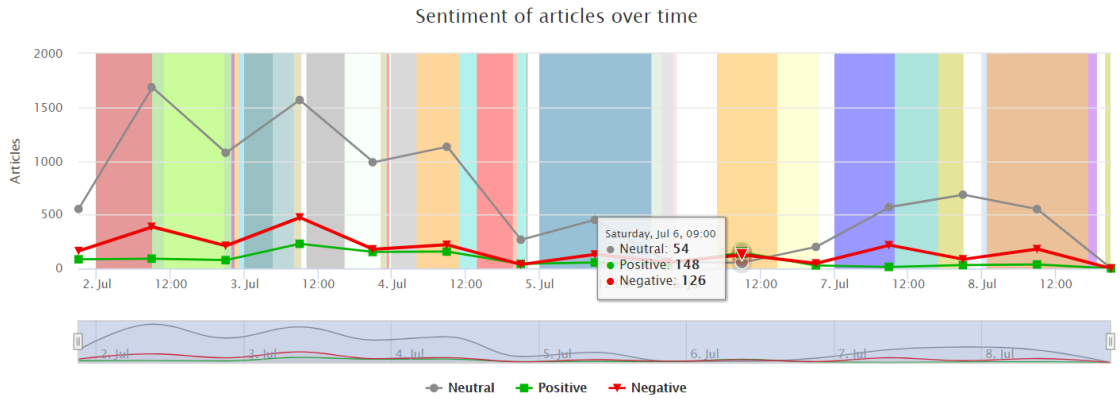

Figure 3: The "sentiment over time" chart for articles, with the colored bands representing events.

3.2. Example Usage Scenario

To better illustrate how the platform is used by real users, we will give an example of a usage scenario. Let's say we have a journalist that wants to investigate what is being said on the social web about the political and economical regularity/normality in Greece. There are three main steps they have to take in order to take advantage of the platform's analytics.

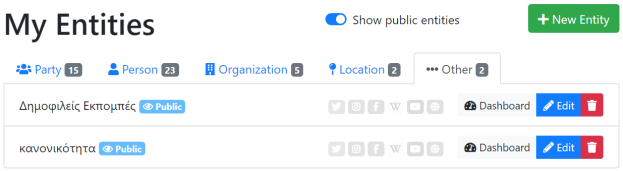

Figure 4: List of the entities defined by users in SWO.

Sign up This step is simple, the journalist simply has to create an account on our platform by providing a username, password and e-mail address. Or, they can use their academic login from many Greek academic institutions.

Define a dashboard by creating a new entity, with any keywords that may be related to it. An example of this screen is seen in Figure 1 with our new entity being defined with a name, a few keywords and two hashtags. After creating the entity, it will be visible in the entities list of the application, shown in Figure 4, with a "Dashboard" button next to it, which will take the user to the dashboard that has been created for that entity.

View analytics By the next morning, a dashboard with analytics will be populated for each created entity. In Figure 2 we see the "Articles" tab of our entity's dashboard with some basic statistics, the "sentiment over time" chart as well as the entity interest chart (how many of the total gathered items reference the entity). Looking at the "Average Polarity" metric we see that the overall sentiment about our entity in articles is positive, as well as its evolution over time in the "sentiment over time" chart. There is also a bar chart, not visible in the figure, showing the top domains which refer to the entity. The user can view the individual items that make up the chart long with their sentiment by clicking on a timepoint of the "sentiment over time" chart (Figure 5). These items are clickable, meaning that the user is able to visit the original sources from which SWO generated its analytics. Finally, in the line chart at the bottom of Figure 2 we can see that the amount of articles mentioning this entity compared to the total gathered articles is almost the same. This means that the interest for our entity in the media also remained the same over time. Similar charts for comments and tweets can be found in the "Comments" and "Tweets" tabs respectively.

To summarize, using the tools available in the Social Web Observatory, journalists can define their own entities, that range from persons to even abstract concepts. Taking sentiment into account, we use various widgets and visualizations in order to present what is being said about the entity. We also display any relevant detected events which help by providing context to the sentiment towards the entity. In the next section we will describe how these events are detected using the articles about an entity.

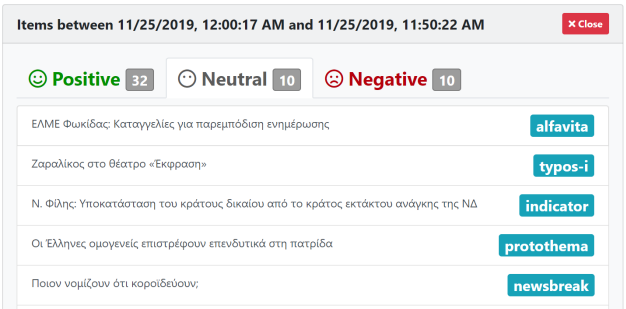

Figure 5: Individual items that make up the sentiment of an entity at a given time.

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. alternative 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. A general architecture diagram of the pipeline is found in Figure 6. We start with the news gathering by crawling a custom list of over 2000 RSS feeds one by one, adding any new articles we find to the Elasticsearch index where we keep all the articles. This process is run every 20 minutes on our server.

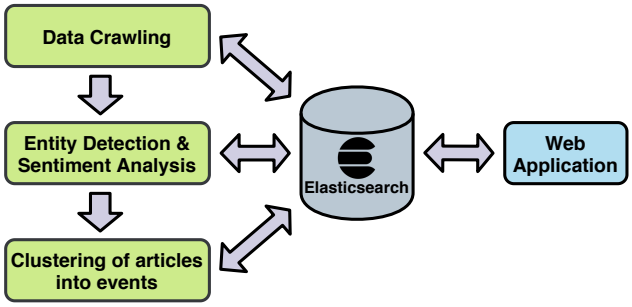

Figure 6: Architecture diagram of the SWO processing pipeline.

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 documents with the entities that they mention, the overall sentiment of each document and the sentiment for each of the entities (calculated by summing the sentiment for each of the entity's occurrences).

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. However, looking at the articles that we gather we see that most of them are published between 9 AM to around 9 PM, so the separation of events by day should not be a problem. 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. We compare the representations with each other to calculate the similarity matrix. 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 Elasticsearch index for events.

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 3. 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)
- Which articles do they feel are irrelevant to others? (List of irrelevant articles)
- Does the cluster (event) title reflect the event well? (Badly/Barely Acceptably/Well enough)

We asked our annotators to work on two sets of data containing 30 events each from different time periods. The 30 events in each set were sampled randomly from the 150 events with the most news articles in the time period. The first time period was July 1-14 of 2019, where in Greece elections happened, and the second was September 7-15 of 2019, a week that the 84th Thessaloniki International Fair was happening, an event that gathered quite a bit of attention. 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. We also uploaded the code for performing the evaluation and converting the annotator's answer to CSV format (used to perform our analysis) to a public repository[2].

[2] https://github.com/leots/swo-events-evaluation

Annotator Pair	Elections p-value	84th T.I.F. p-value
G & K	0.326	0.601
G & O	0.161	0.442
K & O	0.17	0.147

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

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 three different tests. First we looked at their answers on whether they felt that the cluster's articles represented a single event, to see if there are any differences there. Second, 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. Finally, to see if the annotators agree on which articles are irrelevant in each event, We also calculated the Jaccard similarities between the pairs of lists and got the mean Jaccard similarity for each annotator pair.

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, 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 looked at their answers for whether each cluster seems to represent a single event. In all events of both datasets, the answer was always yes, except in one case in the Elections dataset where one of the annotators answered no. From this we can conclude that the annotators agree that each cluster represents a single event in the vast majority of cases.

We also performed a set of paired t-tests between the annotators to show if the distributions of errors (number of 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) for both datasets. This means that the annotators seem to agree on how many articles are irrelevant in each cluster, which indicates a consistent evaluation process.

For the last question, we compared the lists of irrelevant articles that each annotator found, for all annotator pairs. As we can see in Table 2 the mean Jaccard similarity be-

Annotator Pair	Elections	84th T.I.F.
G & K	0.911 ± 0.05	0.833 ± 0.07
G & O	0.933 ± 0.05	0.933 ± 0.05
K & O	0.933 ± 0.05	0.9 ± 0.06

Table 2: Mean Jaccard similarity of the irrelevant articles identified by each annotator pair in each cluster.

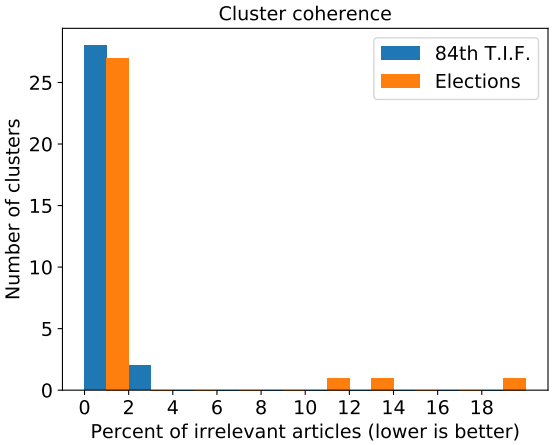

Figure 7: Clustering coherence according to the annotators.

tween the lists of irrelevant articles for each annotator pair and dataset was very high, in most cases more than 90%. 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 7) 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. For the 84th T.I.F. conference, we see that most clusters contain less than 2% irrelevant articles, while the rest still don't have more than 4% of irrelevant articles. In the Elections dataset we still have most clusters containing less than 2% irrelevant articles, and a few more with a higher percentage but still never more than 20%. 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 for the Elections dataset).

We next studied the cluster size distribution to better understand if the clusters were also useful (i.e. non-trivial, having only 1 article). Looking at the cluster size statistical summary (quartiles) for each dataset in Table 3, we see that the minimum number of articles found in any cluster for the Elections dataset is 3, and for the 84th T.I.F. it is 24. In general, the events for the 84th T.I.F. dataset contain many more articles than the Elections one, but in both datasets the clusters are non-trivial. Therefore, we can draw the conclusion that the clusters seem to be coherent, meaningful and useful.

Dataset	Min	1st Qu.	Median	3rd Qu.	Max
Election	3	3.25	5	8.75	127
84th T.I.F.	24	34	49	62	127

Table 3: Basic statistical summary of cluster sizes for each dataset.

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.

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 the Elections dataset, in 22 of the 30 events (73% of the cases) the quality was at least 1 (acceptable) on average. In 26% of the events the title was considered good enough. In the 84th T.I.F. dataset the titles seemed to be even better, with 26 out of 30 being at least acceptable, while 40% of the titles were good enough. We can conclude that the users seem to be able to understand what events are about from their title.

In the final section we will summarize what we did in this work 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 backend system which analyzes the gathered data. We described the processes within the platform as well as its available functionalities in detail. Part of this system is using event detection to show events to the user, in order to help them explain the sentiment about an entity 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 most 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. Finally, we would like to include even more tools and analytics to improve the available functionality of the platform.

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).

8. Bibliographical References

Beineke, P., Hastie, T., Manning, C., and Vaithyanathan, S. (2004). Exploring sentiment summarization. In *Proceedings of the AAAI spring symposium on exploring attitude and affect in text: theories and applications*, volume 39. The AAAI Press Palo Alto, CA.

Feng, X., Qin, B., and Liu, T. (2018). A language-independent neural network for event detection. *Science China Information Sciences*, 61(9), September.

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

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

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

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

Hasan, M., Orgun, M. A., and Schwitter, R. (2018). A survey on real-time event detection from the Twitter data stream. *Journal of Information Science*, 44(4):443–463, August.

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., and Zhu, Q. (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*, pages 1127–1136. Association for Computational Linguistics.

Imran, M., Castillo, C., Diaz, F., and Vieweg, S. (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*, pages 507–511, Lyon, France. ACM Press.

Klenner, M., Petrakis, S., and Fahrni, A. (2009). Robust compositional polarity classification. In *Proceedings of the International Conference RANLP-2009*, pages 180–184, Borovets, Bulgaria, September. Association for Computational Linguistics.

Leban, G., Fortuna, B., Brank, J., and Grobelnik, M. (2014). Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*, pages 107–110, Seoul, Korea. ACM Press.

Lerman, K., Blair-Goldensohn, S., and McDonald, R. (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*, pages 514–522, Athens, Greece. Association for Computational Linguistics.

Litvak, M., Vanetik, N., Levi, E., and Roistacher, M. (2016). What's up on twitter? catch up with twist! In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 213–217.

Montalvo, S., Martínez, R., Fresno, V., and Delgado, A. (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, February.

Papachristopoulos, L., Ampatzoglou, P., Seferli, I., Zafeiropoulou, A., and Petasis, G. (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, May.

Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I., and Spyropoulos, C. D. (2002). Ellogon: A New Text Engineering Platform. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 72–78, Las Palmas, Canary Islands, Spain, May 29–31. European Language Resources Association.

Petasis, G., Spiliotopoulos, D., Tsirakis, N., and Tsantilas, P. (2014). Sentiment analysis for reputation management: Mining the greek web. In Aristidis Likas, et al., editors, *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, May 15-17, 2014. Proceedings*, volume 8445 of *Lecture Notes in Computer Science*, pages 327–340. Springer.

Pittaras, N., Giannakopoulos, G., Tsekouras, L., and Varlamis, I. (2018). Document clustering as a record linkage problem. In *Proceedings of the ACM Symposium on Document Engineering 2018*, page 39. ACM.

Toda, H. and Kataoka, R. (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*, pages 81–86. ACM.

Tsekouras, L., Varlamis, I., and Giannakopoulos, G. (2017). A graph-based text similarity measure that employs named entity information. In *RANLP*, pages 765–771.

Tsekouras, L., Petasis, G., and Kosmopoulos, A. (2019). Social web observatory: An entity-driven, holistic information summarization platform across sources. In *Proceedings of MultiLing 2019 Workshop, RANLP 2019, Varna, Bulgaria, September 2-4, 2019*.

Weng, J. and Lee, B.-S. (2011). Event detection in twitter. In *Fifth international AAAI conference on weblogs and social media*.