Event Registry – Learning About World Events From News

Gregor Leban, Blaž Fortuna, Janez Brank, Marko Grobelnik
Jožef Stefan Institute
Ljubljana, Slovenia
name.surname@ijs.si

ABSTRACT

Event Registry is a system that can analyze news articles and identify in them mentioned world events. The system is able to identify groups of articles that describe the same event. It can identify groups of articles in different languages that describe the same event and represent them as a single event. From articles in each event it can then extract event's core information, such as event location, date, who is involved and what is it about. Extracted information is stored in a database. A user interface is available that allows users to search for events using extensive search options, to visualize and aggregate the search results, to inspect individual events and to identify related events.

Keywords

information extraction; named entity detection; cross-linguality; clustering; news

1. INTRODUCTION

There are thousands of news articles written and published every day by news agencies all across the world. They are written in various languages and discuss all possible topics. A large percentage of these articles are discussing world events - current, past and future. There is no generally accepted definition of an event, but one intuitive definition is that an event is any significant happening in the world. Two instances of an event are, for example, Felix Baumgartner's jump from a helium balloon on October 14, 2012 and bombings during the Boston marathon on April 15, 2013.

The way people today learn about present and past world events leaves much to be desired. Firstly, despite being interested in events, the actual "unit" of content that we consume is a news article. Although all articles describing events answer the main questions about who, where, when and what, this information remains hidden in the text and requires the reader to manually extract it by reading the article. Reading a single article can also give a biased and uncomplete picture of the event which is why it is good to find related articles from other news publishers – a process that, of course, has to be done manually. Searching for events or related events is also problematic since it mostly relies on searching using one or more relevant keywords.

To make learning about the events easier we present in this paper a system called Event Registry. It is able to collect news articles from thousands of news sources and identify in them the events that are being discussed. Information about the events is automatically extracted from the articles and stored in a database. The events can then be found by specifying a search condition such as an entity, topic, location or date. Events matching the criteria can listed as well as summarized and visualized in different ways in order to provide additional insights. For each event, individual articles describing the event can be viewed, as well as related events.

The rest of the paper is organized as follows. In the next section we describe the architecture of the system with brief details about individual parts of the pipeline. Afterwards, we describe the features of the web interface that can be used for finding and visualizing events.

2. EVENT REGISTRY ARCHITECTURE

The presented system for detecting world events consists of a set of components that are illustrated in Figure 1. The pipeline contains four main parts: (a) data collection, which is responsible for collecting news articles, (b) pre-processing steps, where we annotate and extract information from individual articles, (c) event construction, where we group articles describing the same event and extract event information, and (d) event storage, where we store events and provide methods for accessing them. Each part will be now briefly described.

Data collection

For collecting data we use News Feed service [7] which collects news articles from around 75.000 news sources. The number of collected articles ranges between 100.000 and 200.000 articles per day. The collected articles are in various languages, where most represented languages are English (50% of all articles), German (10%), Spanish (8%) and Chinese (5%). These languages are also the only ones that we syntactically and semantically process in the following steps of the pipeline.

Pre-processing steps

The articles in the mentioned languages are then processed with a set of linguistic tools. A very important component for Event Registry is the named entity recognizer which detects the named entities mentioned in the articles

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577024 .

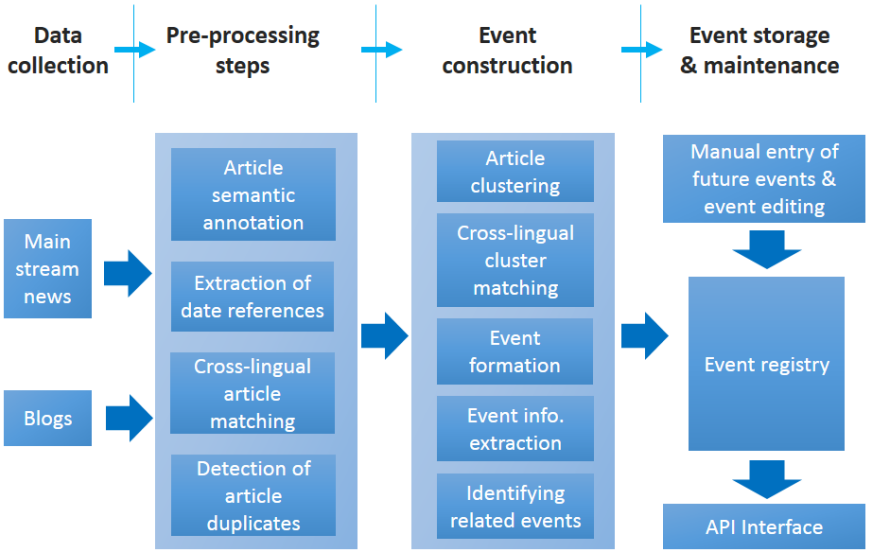

Figure 1: Pipeline used for the Event Registry.

and disambiguates them. Since events are associated with a date we also try to identify in the text date mentions using a set of regular expressions for different languages.

An important functionality of Event Registry is finding groups of articles describing the same event, no matter in which language the articles are written. To support this functionality we combine different learning features, where one of them is cross-lingual similarity of articles. The cross-lingual similarity service[6] can compute an approximate similarity between articles written in English, German, Spanish and Chinese language. The computation is based on an aligned set of basis vectors obtained using latent semantic indexing and a generalized version of canonical correlation analysis. As an output, the service can provide for each article a set of recent most similar articles in other languages and an approximate similarity score.

Event construction

In the event construction phase we are trying to find groups of articles describing the same event and extract from the articles event information. In order to identify groups of articles describing the same event we implemented an online clustering algorithm based on [4, 3]. We use a separate clustering instance for each of the four languages. Each article is represented using the vector space model based on the article title, body and detected named entities (entities are assigned much higher weights than ordinary words). Based on the computed vector, each new article is put into the closest cluster. After every n added articles we reevaluate the clusters and check if some clusters need to be merged or split into two. Bayesian information criterion is used when deciding if two clusters should be split into two. Alternatively, the cosine similarity and the Lughofer's ellipsoid criterion[5] are used to decide if two clusters are similar enough to be merged. Since articles on the same event are typically reported only for a few days, we delete the clusters (only from clustering, not from the Event Registry) that contain articles that are more than k days old.

As a result of the clustering we get groups of articles that describe the same event in a single language. In order to group clusters about the same event in different languages, we use an SVM model. The learning data that we used for building the model consists of pairs of clusters, for which experts manually decided if they describe the same event or not. For each pair of clusters we extract various features that are relevant in deciding if two clusters discuss the same event or not. One of the main features is the cross-lingual cluster similarity that is computed based on cross-lingual article similarities. For tested clusters C_1 and C_2 we check for each article $a_i \in C_1$, how many of its most similar articles are in cluster C_2, and vice-versa. Another important feature is the similarity of most frequently annotated entities in both clusters. Since named entities are represented with language independent identifiers we can directly compare clusters based on entities, regardless of cluster language. Other learning features include also the time difference between the clusters, time variability inside the clusters, cluster qualities, etc. Given the positive and negative learning examples, the SVM model can predict for a new pair of clusters if they should be merged or not.

Event storage

Once one or more clusters are identified that are believed to belong to the same event, we create an event in the Event Registry and assign it a unique id. To extract event information we analyze the articles in the event's clusters. Event title and a short text snippet are determined by finding the article closest to the center of the cluster (medoid article) and using it's title and first paragraph. For the event date we analyze the detected date references in the articles. If the most frequently detected date is frequent enough then we use it as the event date. If no date passes the threshold then we use the average date of the article in the cluster as the event date. The average date is used instead of the earliest article date to compensate for clustering errors – we don't want an older, incorrectly assigned article to be responsible for incorrectly assigning the event date. To set the event location

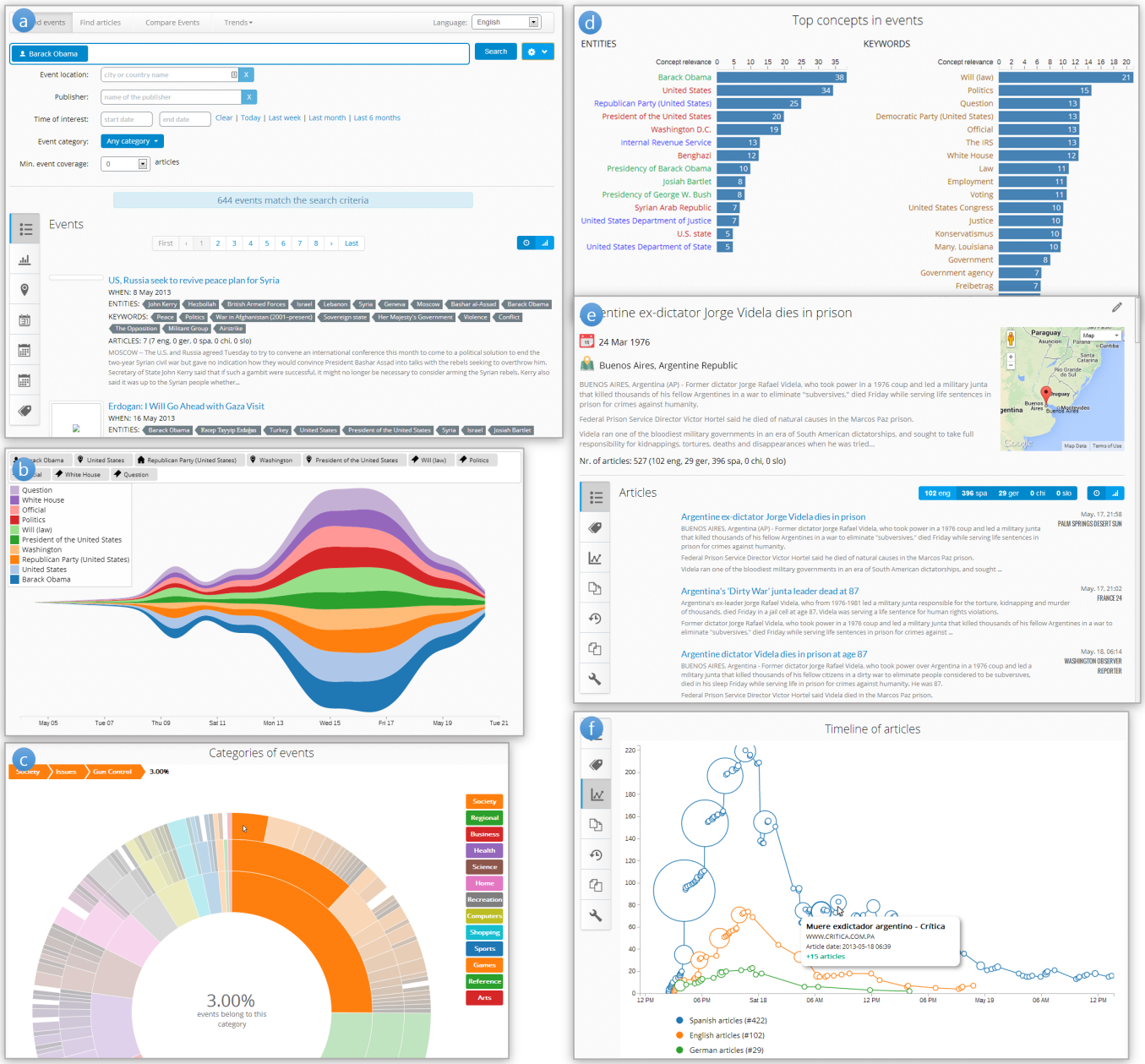

Figure 2: Screenshots of the Event Registry user interface.

we find frequently detected named entities that are known to be locations (based on GeoNames[2]). Especially high weights are put on the locations that appear at the start of the articles. The location with the highest weight is chosen as the event location. As a way of summarizing what the event is about we analyze all detected named entities in the articles and compute their weight based on how commonly they appear in the articles. Events are also about different topics (sports event vs. bombing report). To categorize the events we used the DMoz taxonomy[1] which contains a categorization of 5 million web pages. We built a DMoz classifier that can classify each event into a DMoz category based on the content of the articles in the event.

Events with all the extracted information about them are then stored in the Event Registry database and are searchable using the search API.

3. USER INTERFACE

The Event Registry demo[1] contains around 28.000 events that were extracted from 420.000 news articles from a 14 day period in mid May 2013. A screencast is available at [2].

3.1 Search options

[1]http://www.eventregistry.org
[2]http://www.eventregistry.org/intro

The search interface allows the users to search for events based on different criteria (see Figure 2.a). The main input box is an autocomplete field where the user can specify one or more concepts (entities or keywords) of interest. Only events that are associated with all specified concepts will be shown as search results. The user can also specify as a condition the event location, time of interest, event category and the minimum event coverage.

3.2 Displaying search results

After performing the search it is not uncommon to find thousand or more events that match the criteria. In order for the user to better understand the results and possibly refine the query we provide different ways of presenting the results. The most common way is to check the list of events, where we show for each event the main extracted information. The list of events can be sorted either by date or by relevance to the query.

For getting the big picture of the search results we can generate a number of visualizations. The concept visualization (Figure 2.d) displays a bar chart containing the most relevant entities and keywords discussed in the events. Each concept is also associated with a relevance score that describes the average relevance of the concept (on the 0 – 100 scale) in the resulting events. The location visualization displays the map of locations where the events occur. The timeline visualization shows the distribution of the events over time. The trending concepts graph (Figure 2.b) uses the themeriver visualization to show how the popularity of top concepts changes in the events over time. The visualization should be especially useful when viewing events ranging over a longer time period when themes actually do change significantly. To understand the co-occurrence of concepts in the events, the entity graph can be used. It displays a network of top entities in the results, where edges are drawn between the entities that frequently co-occur in the same events. Lastly, the visualization of categories (Figure 2.c) shows the categorization of events using the DMoz taxonomy.

3.3 Displaying event information

Clicking an individual event in the event list opens the event in a separate window. An example of such a window is shown in Figure 2.e. Top part of the window shows the title, location, date and a short summary of the event. Below is a list of articles describing the event. The articles are grouped by language and a particular language can be selected by clicking the appropriate button. As it can be seen in the example, the system was able to automatically identify and merge articles reporting about the same event in three different languages. By clicking the title of an article, the actual content of the article can be seen.

In order to quickly understand what the event is about, the concept visualization displays top entities and keywords for the event. To see the trending properties of the event, the article timeline visualization (Figure 2.f) displays time when the articles about the event were written. The height of the curve indicates the cumulative number of articles about the event in the last 6 hours. The size of the point indicates the number of articles that were reported at the same moment.

An important feature when viewing an event is also the ability to display related events. Related events are found by computing the TF-IDF weights on the event concepts and finding other events with similar concept weights (by using cosine similarity measure). The similar events can be shown in two ways – as a bar chart of events, order by decreasing similarity, or on a timeline where the order of events is defined by event time. In any case, the related events can help the user to expand from a single event and to maybe understand what were the events leading up to it and what were the consequences of it.

4. FUTURE WORK

The Event Registry is already able to extract a lot of information about an event. We would however like to extend this information also with information about the relations between the entities relevant for the event. In case of Barack Obama meeting with David Cameron we would like to understand that the relation between these two entities was "to meet". In this way we would be able to generate semantic graphs with entities as nodes and relations as edges of the graph. As a next step, we would like to identify slots for different event types (e.g. football match, meeting, earthquake, ...) and try to automatically populate them from using the articles. For example, in the case of an earthquake event, we would expect to populate slots such as the earthquake location, magnitude, the number of casualties, etc.

We also plan to analyze the similarities and differences between the articles describing the same event. We would like to detect how information is spread and copied between different publishers and detect how sentiment varies depending on the news source.

Research plans for the future also include identifying cause and effect relations between the detected events. By finding examples of such events we would like to generalize information in them and use it to predict what are the potential effects a new event might have.

5. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and X-Like (ICT-288342-STREP).

6. REFERENCES

[1] DMoz, open directory project, *http://www.dmoz.org/*.

[2] GeoNames, *http://www.geonames.org/*.

[3] C. C. Aggarwal and S. Y. Philip. On clustering massive text and categorical data streams. *Knowledge and information systems*, 24(2):171–196, 2010.

[4] C. C. Aggarwal and P. Yu. A framework for clustering massive text and categorical data streams. In *Proceedings of the sixth SIAM international conference on data mining*, volume 124, pages 479–483, 2006.

[5] E. Lughofer. A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*, 3(3):135–151, 2012.

[6] J. Rupnik, A. Muhic, and P. Skraba. Cross-lingual document retrieval through hub languages. In *NIPS*, 2012.

[7] M. Trampus and B. Novak. Internals of an aggregated web news feed. In *Proceedings of 15th Multiconference on Information Society 2012 (IS-2012)*, 2012.