Cross-lingual detection of world events from news articles

Gregor Leban, Blaž Fortuna, Janez Brank, and Marko Grobelnik

Jožef Stefan Institute
Ljubljana, Slovenia
{firstname.surname}@ijs.si

Abstract. In this demo we describe a system called Event Registry (*http://eventregistry.org*) that can identify world events from news articles. Events can be detected in different languages. For each event, the system can extract core event information and store it in a structured form that allows advanced search options. Numerous visualizations are provided for visualizing search results.

Keywords: information extraction; cross-linguality; events; named entity detection; clustering; news

1 Introduction

Each day there are thousands of small, medium and large events that are happening in the world. These events range from insignificant meetings, conferences and sport events to important natural disasters and decisions made by world leaders. The way we learn about these events is from different media channels. The unstructured nature of content provided on these channels is suitable for humans, but hard to understand by computers.

Identifying events described in the news and extracting main information about these events is the main goal of the system presented in this paper. Event Registry [2] is able to process news articles published in different languages worldwide. By analyzing the articles it can identify the mentioned events and extract main event information. Extracted event information is stored in a structured way that provides unique features such as searching for events by date, location, entity, topic and other event properties. Beside finding events of interest, Event Registry also provides user interface with numerous visualizations showing date, location, concept and topic aggregates of events that match the search criteria.

The rest of the paper is organized as follows. We start by describing the high level architecture of the system. Next, we describe in more details the process of identification of events from individual news articles. We continue by providing information about how event information is extracted from a group of articles mentioning the event. In the end we also describe the main features of the frontend interface of the Event Registry.

2 System architecture

Event Registry consists of a pipeline of components, that each provide unique and relevant features for the system. In order to identify events we first need data from which we can extract the information. In Event Registry we use news articles as the data source. The news articles are collected using the News-Feed [5] service which collects news articles from more than 75.000 worldwide news sources. Collected articles are in more than 40 languages, where articles in English, Spanish, German and Chinese languages amount to about 70% of all articles. These languages are also the only ones we use in the Event Registry.

Each collected article in the mentioned languages is then analyzed in order to extract relevant information. One of the key tasks is identification and disambiguation of named entities and topics mentioned in the article. We perform this task using a semantic enrichment service developed in the XLike project. We also detect date mentions in the text, since they frequently specify the date when the event occurred. By analyzing the articles we noticed that different news sources frequently publish almost identical news articles. These duplicated articles don't bring any new information, which is why we identify them and ignore them in the rest of the pipeline.

An important feature of the Event Registry is cross-linguality. To support it, we identify for each article a set of most similar articles in other languages. To identify these articles we use canonical correlation analysis[4] which maps articles from different languages into a common semantic space. The common space can then be used to identify most similar articles in all other languages. In order to train the mapping to the common space we used the aligned corpus of Wikipedia articles.

After extracting relevant features from each individual article we start with the main task of identifying events from groups of articles. Since this is the main contribution of the paper we will describe the details of the process in the next three sections.

3 Identification of events

An assumption that we make in identifying events is that any relevant event should be reported at least by a few different news publishers. In order to identify events we therefore apply an online clustering algorithm based on [1] on articles as they are added into the system. Each article is first transformed into a TF-IDF weighted feature vector. The features in the vector are the words in the document as well as the identified named entities and topics. If the cosine similarity of the closest centroid is above a threshold, the article is added to the closest cluster and the cluster properties are updated. Otherwise, a new cluster is formed that contains only the new article.

Each identified cluster of articles is considered to describe an event if it contains at least a minimum number of articles (the minimum value used in our system is 5 articles). Once a cluster reaches this limit, we treat it as an event and the information about it's articles are sent to the next components in the pipeline. Those components are responsible for extracting event information from the articles and will be described in the next sections.

Since clusters are being constantly updated with new articles we want to reevaluate each cluster after a few updates in order to determine if it should be split into two clusters or merged with another cluster. In order to decide if the cluster should be split we apply a bisecting k-means algorithm (with $k = 2$) on the cluster. We then use a variant of the Bayesian Information Criterion to decide whether to accept the new split or not. Periodically we also identify pairs of clusters with high similarity and decide if they should be merged or not. The decision is made using the Lughofer's ellipsoid criterion [3]. We assume that the clusters that have not been modified for a few days mention past events and we therefore remove them from the list of maintained clusters.

3.1 Cross-lingual merging of clusters

Each identified cluster of articles contains only articles in a single language. Since articles in different languages can describe the same events we want to identify clusters describing the same event in different languages and represent them as a single event. In order to determine which cluster pairs to merge we represent the task as a binary classification problem. Given a cluster pair c_1 and c_2 in languages l_1 and l_2 we want to extract a set of features that would discriminate between cluster pairs that describe the same event and those that don't. A very important learning feature that we can extract for each cluster pair is computed by inspecting individual articles in each cluster. Using canonical correlation analysis we are able to obtain for each article in c_1 a list of 10 most similar articles in language l_2. Using this information we can check how many of these articles are in c_2. We can repeat the same computation for articles in c_2. By normalizing the results by the size of the clusters we can obtain a score that should by intuition correlate with similarity of the two clusters across the two languages. Some of the other features that we extract include the time difference between the average article date of each cluster, cosine similarity of the annotated concepts and topics, and category similarity.

In order to build a classification model we collected 85 learning examples. Some cluster pairs were selected randomly and some were selected by users based on their similarity. The features for the selected learning examples were extracted automatically, while the correct class was assigned manually. A linear SVM was then trained on the data which achieved 87% accuracy using 10-fold cross validation.

4 Event information extraction

Once we obtain one or more new clusters that are believed to describe a single event, we assign them a new id in the Event Registry. From the associated articles we then try to extract the relevant information about the event. We try to determine event date by seeing if there is a common date reference frequently mentioned in the articles. If no date is mentioned frequently enough we use the average article's published date as the event date. The location of the event is determined by locating frequently mentioned named entities that represent locations. To determine what the event is about we aggregate the named entities and topics identified in the articles. Each event is also categorized using a DMoz taxonomy. All extracted information is stored in the Event Registry in a structured form that provides rich search and visualization capabilities.

5 Event search, visualization options and data accessibility

Event Registry is available at *http://eventregistry.org* and currently contains 15.000.000 articles from which we identified about 1 million events. Available search options include search by relevant named entities, keywords, publishers, event location, date and category. The resulting events that match the criteria can then be seen as a list or using one of numerous visualizations. Main visualizations of search results include location and time aggregates, list of top named entities and topics, graph of related entities, concept trends, concept matrix, date mentions, clusters of events and event categories. For each individual event we can provide the list of articles describing it as well as visualizations of concepts, article timeline, date mentions, article sources and other similar events. Examples of these visualizations are (due to space limit) available on *http://eventregistry.org/screens*. All Event Registry data is also stored using the Storyline ontology and is available through a SPARQL endpoint available at *http://eventregistry.org/sparql*.

6 Acknowledgments

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

References

1. 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.
2. G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event registry – learning about world events from news. In *WWW 2014 Proceedings*, pages 107–110. ACM, 2014.
3. E. Lughofer. A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*, 3(3):135–151, 2012.
4. J. Rupnik, A. Muhic, and P. Skraba. Cross-lingual document retrieval through hub languages. In *NIPS*, 2012.
5. M. Trampus and B. Novak. Internals of an aggregated web news feed. In *Proceedings of 15th Multiconference on Information Society 2012 (IS-2012)*, 2012.