Online-Monitoring of Security-Related Events

Martin Atkinson, Jakub Piskorski, Bruno Pouliquen
Ralf Steinberger, Hristo Tanev, Vanni Zavarella
Joint Research Centre of the European Commission
Institute for the Protection and Security of the Citizen
Via Fermi 2749, 21027 Ispra (VA), Italy
firstname.lastname@jrc.it

Abstract

This paper presents a fully operational real-time event extraction system which is capable of accurately and efficiently extracting violent and natural disaster events from vast amount of online news articles per day in different languages. Due to the requirement that the system must be multilingual and easily extendable, it is based on a shallow linguistic analysis. The event extraction results can be viewed on a publicly accessible website.

1 Introduction

Gathering information about violent and natural disaster events from online news is of paramount importance to better understand conflicts and to develop global monitoring systems for the automatic detection of precursors for threats in the fields of conflict and health. This paper reports on a fully operational live event extraction system to detect information on violent events and natural disasters in large multilingual collections of online news articles collected by the news aggregation system Europe Media Monitor (Best et al., 2005), http://press.jrc.it/overview.html.

Although a considerable amount of work on the automatic extraction of events has been reported, it still appears to be a lesser studied area in comparison to the somewhat easier tasks of named-entity and relation extraction. Two comprehensive examples of the current functionality and capabilities of event extraction technology dealing with the identification of disease outbreaks and conflict incidents are given in (Grishman et al., 2002) and (King and Lowe, 2003) respectively. The most recent trends and developments in this area are reported in (Ashish et al., 2006)

In order to be capable of processing vast amounts of textual data in real time (as in the case of EMM) we follow a linguistically lightweight approach and exploit clustered news at various processing stages (pattern learning, information fusion, geo-tagging, etc.). Consequently, only a tiny fraction of each text is analysed. In a nutshell, our system deploys simple 1 and 2-slot extraction patterns to identify event-relevant entities. These patterns are semi-automatically acquired in a bootstrapping manner by using clustered news data. Next, information about events scattered over different documents is integrated by applying voting heuristics. The results of the core event extraction system are integrated into a real-world global monitoring system. Although we mainly cover the security domain, the techniques deployed in our system can be applied to other domains, such as for instance tracking business-related events for risk assessment.

In the remaining part of this paper we give a brief overview of the real-time event extraction processing chain and describe the particularities of selected subcomponents. Finally, the online application is presented.

2 Real-time Event Extraction Process

The real-time event extraction processing chain is depicted in Figure 1. First, news articles are gathered by dedicated software for electronic media monitoring, namely the EMM system (Best et al., 2005). EMM receives an average of 50,000 news articles per day from about 1,500 news sources in

Coling 2008: Companion volume – Posters and Demonstrations, pages 145–148
Manchester, August 2008

over 40 languages, and regularly checks for updates of news. Secondly, the input data is grouped into news clusters ideally including documents on one topic or event. Then, clusters describing security-related events are selected using keyword-based heuristics. For each such cluster, the system tries to detect and extract only the main event by analysing all documents in the cluster.

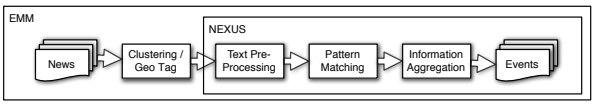

Figure 1: Real-time processing chain.

Next, each cluster is processed by our core event extraction engine. For each detected violent event, it produces a frame, whose main slots are: date and location, number of killed, injured or kidnapped people, actors, type of event, weapons used, etc. In an initial step, each document in the cluster is linguistically pre-processed in order to produce a more abstract representation of the texts. This encompasses: fine-grained tokenisation, sentence splitting, matching of known named entities, labelling of key terms and phrases like action words (e.g. *kill*, *shoot*) and person groups.

Once texts are grouped into clusters and linguistically pre-processed, the pattern engine applies a cascade of extraction grammars (consisting of 1 and 2-slot extraction patterns) on each document within a cluster. For creating extraction patterns, we apply a blend of machine learning and knowledge-based techniques. The extraction patterns are matched against the first sentence and the title of each article from the cluster. By processing only the top sentence and the title, the system is more likely to capture facts about the most important event in the cluster. Even if we fail to detect a single piece of information in one document in a cluster, the same information is likely to be found in another document of the cluster, where it may be expressed in a different way.

Finally, since information about events is scattered over different articles, the last step consists of cross-document cluster-level information fusion, i.e., we aggregate and validate information extracted locally from each single article in the same cluster. For this purpose, simple voting-like heuristics are deployed.

Every ten minutes, EMM clusters the articles found during the last four hours. The event extraction engine analyses each of these clusters. The event information is thus always up-to-date. The output of the event extraction engine constitutes the input for a global monitoring system.

3 Geo-tagging Clusters

Challenges for geo-tagging clusters are that place names can be homographic with person names and with other place names. We solve the former ambiguity by first identifying person names found in our automatically populated database of known people and organisations. For the latter ambiguity, we adopted a cluster-centric approach by weighting all place names found in a cluster and by selecting the one with the highest score. For each cluster, we thus first establish all possible candidate locations by looking up in the texts all place, province, region and country names found in a multilingual gazetteer (including name variants). The weights of the locations are then based on the place name significance (e.g., a capital city scores higher than a village) and on the place name hierarchy (i.e. if the province or region to which the place belongs are also mentioned in the text, it scores higher).

4 Pattern Acquisition

For pattern acquisition, we deploy a weakly supervised bootstrapping algorithm (Tanev and Oezden-Wennerberg, 2008) similar in spirit to the one described in (Yangarber, 2003), which involves some manual validation. Contrary to other approaches, the learning phase exploits the knowledge to which cluster the news items belong. Intuitively, this guarantees better precision of the learned patterns. In particular, for each event-specific semantic role (e.g. *killed*), a separate cycle of learning iterations is executed (usually up to three) in order to learn 1-slot extraction patterns. Each cluster includes articles from different sources about the same news story. Therefore, we assume that each entity appears in the same semantic role (actor, victim, injured) in the context of one cluster. An automatic procedure for syntactic expansion complements the learning. This procedure accepts a manually provided list of words which have identical (or similar) syntactic usage patterns (e.g. *killed*, *assassinated*, *murdered*, etc.). It then generates new patterns from the old ones by substituting for each other the words in the list. After 1-slot patterns are acquired, some of them are used to manually create 2-slot patterns like `X shot Y`.

5 Pattern matching engine

In order to guarantee that massive amounts of textual data can be processed in real time, we have developed ExPRESS (Piskorski, 2007), an efficient extraction pattern engine, which is capable of matching thousands of patterns against MB-sized texts within seconds. The pattern specification language is a blend of two previously introduced IE-oriented grammar formalisms, namely JAPE used in GATE (Cunningham et al., 2000) and XTDL, used in SPROUT (Drożdżyński et al., 2004).

A single pattern is a regular expression over flat feature structures (FS), i.e., non-recursive typed feature structures without structure sharing, where features are string-valued and – unlike in XTDL types – are not organised in a hierarchy. Each such regular expression is associated with a list of FSs which constitute the output specification. Like in XTDL, we deploy variables and functional operators for forming slot values and for establishing contact with the 'outer world'. Further, we adapted JAPEs feature of associating patterns with multiple actions, i.e., producing multiple annotations (possibly nested). An empirical comparison of the run-time behaviour of the new formalism against the other 2 revealed that significant speed-ups can be achieved (at least 30 times faster). ExPRESS comes with a pool of highly efficient core linguistic processing resources (Piskorski, 2008).

6 Information Aggregation

Once single pieces of information are extracted by the pattern engine, they are merged into event descriptions by applying an information aggregation algorithm. This algorithm assumes that each cluster reports at most one main event of interest. It takes as input the text entities extracted from one news cluster with their semantic roles and considers the sentences from which these entities are extracted. If one and the same entity has two roles assigned, a preference is given to the role assigned by the most reliable group of patterns (e.g., 2-slot patterns are more reliable). Another ambiguity which has to be resolved arises from the contradictory information which news sources give about the number of victims. We use an ad-hoc heuristic for computing the most probable estimation for these numbers, i.e., firstly the largest group of numbers which are close to each other is selected and secondly the number closest to the average in that group is chosen. After this estimation is computed, the system discards from each news cluster all the articles whose reported victim numbers significantly differ from the estimated numbers for the whole cluster. Additionally, some victim arithmetic is applied, i.e., a small taxonomy of person classes is used to sum victim numbers (e.g., *gunmen* and *terrorists* belong to the same class of *NonGovernmentalArmedGroup*).

7 Event Classification

After the single pieces of information are assembled into the event description, an event classification is performed. Some of the most used event classes are *Terrorist Attack*, *Bombing*, *Shooting*, *Air Attack*, etc. The classification algorithm uses a blend of keyword matching and domain specific rules. As an example, consider the following domain-specific rule: if the event description includes named entities, which are assigned the semantic role *kidnapped*, as well as entities which are assigned the semantic role *released*, then the type of the event is *Hostage Release*, rather than *Kidnapping*. If the event refers to kidnapped people and at the same time the news articles contain words like *video* or *videotape*, then the event type is *Hostage Video Release*. The second rule has a higher priority, therefore it impedes the *Hostage Release* rule to fire erroneously, when the release of a hostage video is reported.

8 Monitoring Events

The core event extraction engine for English is fully operational since December 2007. There are two online applications running on top of it which allow monitoring events. The first one is a dedicated webpage using the Google Maps JavaScript API (see Figure 2). It is publicly accessible at: `http://press.jrc.it/geo?type=event&format=html&language=en` and provides an instant overview of what is occurring where in the world. A small problem with this application is that it overlays and hides events that are close to each other.

The second application shows the same events using the Google Earth client application. The geo-located data is transmitted via the Keyhole Markup Language (KML) format[1] supported directly by Google Earth.[2] The application is re-

[1] http://code.google.com/apis/kml/documentation/

[2] In order to run it, start Google Earth with KML: http://press.jrc.it/geo?type=event&format=kml&language=en

Figure 2: Event visualisation with Google Maps

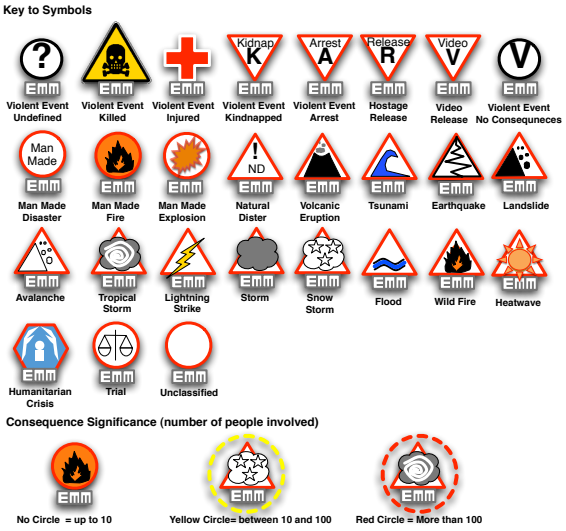

Figure 3: Key to event type icons and magnitude indicators

stricted to displaying at most half the globe, but it allows expanding overlaid events.

Since it is important for stakeholders to be quickly and efficiently informed about the type and gravity of the event, various icons are used to represent the type or group of events visually (see Figure 3). We use general forms of icons for violent events and specific forms of icons for natural and man-made disasters. For violent events, the general form represents the major consequence of the event, except for kidnappings, where specific icons are used. Independently of the type of event, all icons are sized according to the damage caused, i.e. it is dependent on the number of victims involved in the event. Also, to highlight the events with a more significant damage, a border is drawn around the icon to indicate that a threshold of people involved has been passed.

The online demo is available for English, Italian and French. We are currently working on adapting the event extraction engine to other languages, including Russian, Spanish, Polish, German and Arabic. A more thorough description of the system can be found in (Tanev et al., 2008; Piskorski et al., 2008).

References

Ashish, N., D. Appelt, D. Freitag, and D. Zelenko. 2006. *Proceedings of the workshop on Event Extraction and Synthesis, held in conjunction with the AAAI 2006 conference*. Menlo Park, California, USA.

Best, C., E. van der Goot, K. Blackler, T. Garcia, and D. Horby. 2005. Europe Media Monitor. Technical Report EUR 22173 EN, European Commission.

Cunningham, H., D. Maynard, and V. Tablan. 2000. JAPE: a Java Annotation Patterns Engine (Second Edition). Technical Report, CS–00–10, University of Sheffield, Department of Computer Science.

Drożdżyński, W., H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu. 2004. Shallow Processing with Unification and Typed Feature Structures — Foundations and Applications. *Künstliche Intelligenz*, 2004(1):17–23.

Grishman, R., S. Huttunen, and R. Yangarber. 2002. Real-time Event Extraction for Infectious Disease Outbreaks. Proceedings of the Human Language Technology Conference (HLT) 2002.

King, G. and W. Lowe. 2003. An Automated Information Extraction Tool for International Conflict Data with Performance as Good as Human Coders: A Rare Events Evaluation Design. *International Organization*, 57:617–642.

Piskorski, J., H. Tanev, M. Atkinson, and E. Van der Goot. 2008. Cluster-centric Approach to News Event Extraction. In *Proceedings of MISSI 2008*, Wroclaw, Poland.

Piskorski, J. 2007. ExPRESS Extraction Pattern Recognition Engine and Specification Suite. In *Proceedings of the International Workshop Finite-State Methods and Natural language Processing 2007 (FSMNLP'2007)*, Potsdam, Germany.

Piskorski, J. 2008. CORLEONE – Core Linguistic Entity Online Extraction. Technical report 23393 EN, Joint Research Centre of the European Commission, Ispra, Italy.

Tanev, H. and P. Oezden-Wennerberg. 2008. Learning to Populate an Ontology of Violent Events (in print). In Fogelman-Soulie, F. and Perrotta, D. and Piskorski, J. and Steinberger, R., editor, *NATO Security through Science Series: Information and Communication Security*. IOS Press.

Tanev, H., J. Piskorski, and M. Atkinson. 2008. Real-Time News Event Extraction for Global Crisis Monitoring. In *Proceedings of the* 13^{th} *International Conference on Applications of Natural Language to Information Systems (NLDB 2008, Lecture Notes in Computer Science Vol. 5039)*, pages 207–218. Springer-Verlag Berlin Heidelberg.

Yangarber, R. 2003. Counter-Training in Discovery of Semantic Patterns. In *Proceedings of the* 41^{st} *Annual Meeting of the ACL*.