Event Extraction for Balkan Languages

Vanni Zavarella, Dilek Küçük, Hristo Tanev
European Commission
Joint Research Centre
Via E. Fermi 2749
21027 Ispra (VA), Italy
first.last@jrc.ec.europa.eu

Ali Hürriyetoğlu
Center for Language Studies
Radboud University Nijmegen
P.O. Box 9103
NL-6500 HD Nijmegen
a.hurriyetoglu@let.ru.nl

Abstract

We describe a system for real-time detection of security and crisis events from online news in three Balkan languages: Turkish, Romanian and Bulgarian. The system classifies the events according to a fine-grained event type set. It extracts structured information from news reports, by using a blend of keyword matching and finite-state grammars for entity recognition. We apply a multilingual methodology for the development of the system's language resources, based on adaptation of language-independent grammars and on weakly-supervised learning of lexical resources. Detailed performance evaluation proves that the approach is effective in developing real-world semantic processing applications for relatively less-resourced languages.

1 Introduction

We describe a real-time event extraction system for three less-resourced languages: Bulgarian, Romanian and Turkish[1]. The goal of event extraction is to identify instances of a specified set of event types in natural language texts, and to retrieve database-like, structured information about event participants and attributes: these are the entities that are involved in the event and fill type-specific event roles (Ashish et al., 2006). For example, in the fragment *"Three workers were injured in a building collapse"*, the phrase *"three workers"* will be assigned a semantic role `Injured` of the event type `ManMadeDisaster` template.

Gathering and tracking such information over time from electronic news media plays a crucial role for the development of open-source intelligence systems, particularly in the context of global news monitoring of security threats, mass emergencies and disease outbreaks (Yangarber et al., 2005). In this view, it has been proved that being able to rely on highly multilingual text mining tools and language resources is of paramount importance, in order to achieve an unbiased coverage of global news content (Steinberger, 2012).

The system language components include finite state-based entity extraction grammars and domain-specific semantic lexica. These are adapted to the target language from existing language-independent resources or built by using semi-supervised machine learning algorithms, respectively. Most importantly, the lexical acquisition methods we put into place neither make use of any language knowledge nor require to have annotated corpora available.

Section 2 outlines the main processing stages of the application. In Section 3 we describe the methods applied to acquire and adapt the system's language knowledge bases. Finally, in Section 4 we report on an evaluation on event type classification and on the extraction of slot fillers for event templates, and we briefly discuss system performance and prospective improvements.

2 System Architecture

As depicted in Figure 1 (Tanev et al., 2009), first news feeds are clustered, upstream of the event extraction engine, by applying similarity metrics over meta data (named entities, locations, categories) extracted from single articles by dedicated, multilingual software.

Event extraction begins by preprocessing the title and first three sentences of each article within a cluster. This encompasses: fine-grained tokenization, sentence splitting, domain-specific dictionary look-up (i.e. matching of key terms indicating numbers, quantifiers, person titles, per-

[1] While belonging to three distant language families, namely Slavic, Romance and Turkic, respectively, they are spoken in the same geopolitical area, the Balkans.

Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 65–68,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

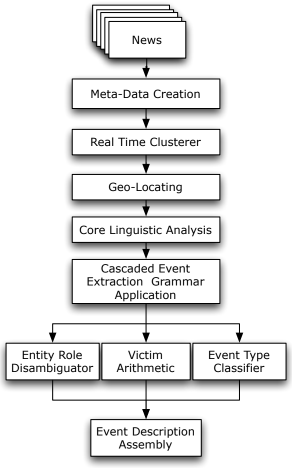


Figure 1: Event extraction processing chain

son groups descriptors like *civilians*, *policemen* and *Shiite*), and finally morphological analysis, simply consisting of lexicon look-up on large domain-independent morphological dictionaries from the MULTEXT project (Erjavec, 2004). Subsequently, a multi-layer cascade of finite-state extraction grammars in the ExPRESS formalism (Piskorski, 2007) is applied on such more abstract representation of the article text, in order to: a)identify entity referring phrases, such as persons, person groups, organizations, weapons, etc. b) assign them to event specific roles by linear combination with event triggering surface patterns. For example, in the text *"Iraqi policemen shot dead an alleged suicide bomber"* the grammar should extract the phrase *"Iraqi policemen"* and assign to it the semantic role `Perpetrator`, while the phrase *"alleged suicide bomber"* should be extracted as `Dead`. We use a "lexicon" of 1/2-slot patterns of the form:

```
<DEAD[Per]> was shot by <PERP>
<KIDNAP[Per]> has been taken hostage
```

where each slot position is assigned an event-specific semantic role and includes a type restriction (e.g. `Person`) on the entity which may fill the slot.

Finally, we aggregate and validate information extracted locally from each single article in the same cluster, such as entity role assignment, victim counts and event type.

We categorize the main event from each cluster with respect to a fine-grained event type set, shown in Table 1.

The event classification module consists of a blend of keyword matching, event role detection and a set of rules controlling their interaction. First, for each event type, we deploy: a) a list of weighted regular expression keyword patterns: each pattern match is awarded the corresponding weight, and an event type is triggered when the weight sum exceeds a defined threshold; b) a set of boolean pattern combinations: `OR` pattern lists are combined by the `AND` operator, each pattern is a restricted regular expression and conjunctions are restricted by proximity constraints. For example in order to detect *TerroristAttack* we use the following combination (translated here in English): ("bomb" `OR` "explosion" `OR`....) `AND` ("terrorist" `OR` "Al Qaida" `OR`..).

Besides the event `TYPE`, the other main slots of an output event frame include: `TYPE`, `DEAD`, `DEAD-COUNT`, `ARRESTED`, `ARRESTED-COUNT`, `PERPETRATOR`, `WEAPON`, etc.

The system will be demonstrated using a KML-aware earth browser[2]. Figure 2 shows a sample output event template.

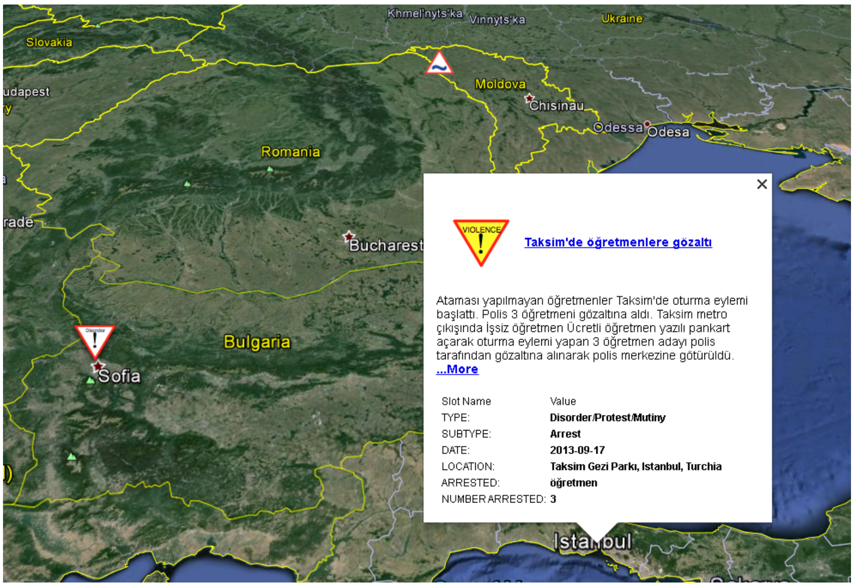
3 Development of language resources

The system's language components are:

Event grammar rules They consist of regular expressions over flat feature structures whose elements include, among the others, semantic types from the domain lexica. We use them to locally parse semantic entities such as person names, person group descriptions, and their clausal combination with verbal event patterns (see Section 2). Grammars in target languages are compiled by adapting the existing rules from source languages, such as English, while the bulk of grammar development mostly consists of providing suitable lexical resources.

Semantic dictionaries Domain-specific lexica, listing a number of (possibly multi-word) expressions sub-categorized into semantic classes relevant for the event domain, with limited or no linguistic annotation, are used by entity recognition grammar rules. Such lexica were created using the weakly supervised terminology extraction algorithm LexiClass (Ontopopulis), described in (Tanev et al., 2009). In order to enforce syntactic

[2] E.g. Google Earth. Notice that Geocoding is currently performed at the level of article text by a language-independent algorithm which is not yet integrated within the event detection process, while Document Creation Date is currently used as the event Date slot filler.



Taksim'de öğretmenlere gözaltı

Ataması yapılmayan öğretmenler Taksim'de oturma eylemi başlattı. Polis 3 öğretmeni gözaltına aldı. Taksim metro çıkışında İşsiz öğretmen Ücretli öğretmen yazılı pankart açarak oturma eylemi yapan 3 öğretmen adayı polis tarafından gözaltına alınarak polis merkezine götürüldü. ...More

Slot Name	Value
TYPE:	**Disorder/Protest/Mutiny**
SUBTYPE:	**Arrest**
DATE:	**2013-09-17**
LOCATION:	**Taksim Gezi Parkı, Istanbul, Turchia**
ARRESTED:	**öğretmen**
NUMBER ARRESTED:	**3**

Figure 2: A sample output template of the system

constraints (e.g. Case) into event clause rules for Romanian language, we have enriched learnt lexical entries for the semantic classes with morphological annotations, using MULTEXT resources. For Turkish, as we do not currently perform morphological analysis, we have rather included common inflected forms of the applicable lexical entries, resulting in larger lexica.

Event triggering patterns They are also acquired semi-automatically, starting with a set of seed examples and an article clustering, by deploying the paraphrase learning algorithm described in (Tanev et al., 2008). For Bulgarian, the grammar, semantic dictionaries and event patterns were created simultaneously, following a semi-automatic approach, described in (Tanev and Steinberger, 2013). In particular, we learned a list of terms referring to people, institutions and organizations and the corresponding pre- and post-modifiers (about 5000 terms). In the same manner, we learned about 550 surface patterns for killing, injuring, kidnapping and arresting actions, together with a 4 level grammar cascade.

Keyword terms The keyword sets used in the event type definitions, namely the OR lists in the boolean pattern combinations (see Section 2 above), can be viewed as instances of some more abstract semantic classes, that a domain expert uses to model a target event scenario. These classes are semi-automatically acquired using the LexiClass algorithm, and then manually combined. As Turkish is an agglutinative language, we have frequently added wildcards at the ends of keywords to cover possible inflected forms.

Table 1: Event type set

AirMissileAttack	Landslide
ArmedConflict	LightningStrike
Arrest	ManMadeDisaster
Assassination	MaritimeAccident
Avalanche	PhysicalAttack
BioChemicalAttack	Robbery
Bombing	Shooting
Disorder/Protest/Mutiny	Stabbing
Earthquake	Storm
Execution	TerroristAttack
Explosion	TropicalStorm
Floods	Tsunami
HeatWave	Vandalism
HeavyWeaponsFire	VolcanicEruption
HostageVideoRelease	Wildfire
HumanitarianCrisis	WinterStorm
Kidnapping	NONE

4 Experiments and Evaluation

System performance is evaluated on three different extractive tasks, carried out on the titles and first three sentences of single news articles: event type classification, event role name/description extraction and victim counting.

We collected test corpora of 52, 126 and 115 news articles for Bulgarian, Romanian and Turkish, respectively, spanning over a time range of 2 months[3]. For each article in the gold standard, we

[3] Articles were manually selected using news aggregators such as Google News. Type distribution resulted in zeroes for

Table 2: System performance in single article extraction mode.

Lang	Type	Dead			Injured			Arrested			Kidnapped			Perpetrator		Weapon	
	MRR	mF	MF	MSE	mF	MF	MSE	mF	MF	MSE	mF	MF	MSE	mF	MF	mF	MF
BG	0.34	0.27	0.68	17.08	0.44	0.6	108.82	0.22	1.0	7.69	0.4	0.5	0.71	0.0	0.0	0.39	1.0
RO	0.22	0.48	0.73	36.53	0.46	0.97	18.57	0.39	0.82	80.5	0.2	1.0	2.14	0.07	0.67	0.1	0.2
TR	0.66	0.73	0.79	16.41	0.85	0.91	0.24	0.31	0.36	52.17	0.4	0.33	0.82	0.25	0.67	0.77	1.0

annotated: a list of applicable types, ordered by relevance, for the main event reported in the article; the set of all the names/descriptions occurring in the text for each applicable event role, merging morphological variants; the cumulative count for the roles Dead, Injured, Kidnapped and Arrested.

Event type classification is evaluated by applying an adapted version of the mean reciprocal rank (MRR) score, used in Information Retrieval to evaluate processes producing a list of relevance ordered query responses. In our case, the MRR for a set of N articles is:

$$MRR = \frac{1}{|N|}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where $rank$ is the rank of the system type response within the gold standard type list for each article.

For each role name/description extraction separately, we compute standard Precision, Recall and F1-measure on system responses, based on partial, n-gram match with gold standard responses, ignoring morphological suffixes.

Finally, we record the root Mean Squared Error (MSE) of system output victim count values against gold standard, over all applicable roles.

Table 2 summarizes the evaluation results. mF and MF columns for each role description task represent respectively the micro and macro average F1-measure over the test set.

Overall, the performance figures are in line with previous evaluations on other languages (Tanev et al., 2009). This proves the methodology is effective on adapting the system to new languages even with little lexical and syntactical proximity. Turkish system consistently outperforms the others, and it also underwent the most resource development cycles: this suggests that applying learning iterations, alternated with human filtering, to the language resources, can increase system accuracy, eventually making it usable for real-world applications. System accuracy is still unreliable for victim counting. One of the main reasons for large errors in victim counting is that the system interprets historical victim statistics reported in articles as event instances. We are currently implementing temporal and discourse heuristics to mitigate this problem.

Acknowledgments

This study is supported in part by a postdoctoral research grant from TÜBİTAK and by the Dutch national program COMMIT.

References

Naveen Ashish, Doug Appelt, Dayne Freitag, and Dmitry Zelenko. 2006. Proceedings of the workshop on event extraction and synthesis. Technical report, AAAI.

Tomaz Erjavec. 2004. MULTEXT-East morphosyntactic specifications.

Jakub Piskorski. 2007. ExPRESS–extraction pattern recognition engine and specification suite. In *Proceedings of the International Workshop Finite-State Methods and Natural language Processing*.

Ralf Steinberger. 2012. A survey of methods to ease the development of highly multilingual text mining applications. *Language Resources and Evaluation*, 46(2):155–176.

Hristo Tanev and Josef Steinberger. 2013. Semi-automatic acquisition of lexical resources and grammars for event extraction in Bulgarian and Czech. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 110–118.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In E. Kapetanios, V. Sugumaran, and M. Spiliopoulou, editors, *Natural Language and Information Systems*, volume 5039 of *Lecture Notes in Computer Science*, pages 207–218.

Hristo Tanev, Vanni Zavarella, Jens Linge, Mijail Kabadjov, Jakub Piskorski, Martin Atkinson, and Ralf Steinberger. 2009. Exploiting Machine Learning Techniques to Build an Event Extraction System for Portuguese and Spanish. *Linguamática*, 1(2):55–66.

Roman Yangarber, Lauri Jokipii, Antti Rauramo, and Silja Huttunen. 2005. Extracting information about outbreaks of infectious epidemics. In *Proceedings of the HLT/EMNLP*.

some less frequent event types.