A Pipeline for Multilingual Protest Event Selection and Annotation

Vera Danilova
Autonomous University of Barcelona,
Spain
Russian Presidential Academy of National Economy and Public Administration,
Russian Federation
Email: vera.danilova@e-campus.uab.cat

***Abstract*—This paper presents a pipeline for multilingual analysis of contentious social behaviour. Its basic functionalities include news articles extraction from a variety of multilingual news sources (in Bulgarian, French, Polish, Russian, Spanish, and Swedish), protest event selection and an ontology-based event annotation. The results are output in CSV format. An evaluation of protest event selection and annotation (Event_Reason slot) algorithms is presented.**

I. Introduction

The history of quantitative protest event analysis dates back to early 1960s. It started with the manual collection of event datasets in order to find reliable patterns of contentious collective behaviour and accumulate statistics for protest prediction and the analysis of its origins, dynamics and aftermath. However, human-based event collection turns out to be subjectivity-prone [10] and insufficient in terms of source, time and location coverage [12]. Even for a single event, it is too time-consuming to perform cross-lingual analysis to complete the picture on its origins, consequences, actors involved, and parallels to other similar events. Since the advent of the Internet, social research institutions, such as Berkman Center for Internet and Society, University of Illinois at Urbana-Champaign Cline Center for Democracy and others, use the world wide ”memory” to improve the time-honoured manual political event data collection. In 1990’s, half-automated systems for political event data collection based on KEDS (Kansas Event Data System[1]) were elaborated that use keyword-based search in digital archives, and domain ontologies and dictionaries. Currently, there are several fully-fledged global projects that deal with political event collection.

Researchers list the following advantages of automatic event data coding: ability to process large datasets in a short span of time, geotemporal coverage, replicability, modifiable dictionaries, no need to pay expensive human coders, objective view of the data. As the drawbacks they mention the inability of the systems to parse complex sentence structures, metaphore, idioms, i.e. figurative language, and timedependent text [10].

In this paper, we present a new pipeline that uses generic multilingual patterns based on the informational structure properties of news headlines to extract protest event data. Currently, it handles lead sentences in Bulgarian, French, Polish, Russian, Spanish, and Swedish, however, more languages can be added. The annotation relies on an in-house experimental ontology, constructed by a domain expert on the basis of a manual analysis of several thousand headlines. This pipeline can be used independently or integrated, together with the ontology, into a larger system for monitoring and analysis of protest events. In [4], we have proved that the extracted event scenario data enhances short-text (headlines) clustering accuracy.

The rest of the paper is organized as follows: Section 2 gives a short overview of CAMEO and SSP ontologies-based extraction systems, as well as of the two early prototypes by [5] and [12]. Section 3 gives details on our generic pipeline, and Section 4 presents the evaluation of event selection and annotation. Section 5 outlines the conclusions and future work.

II. Related Work

For the purposes of our work, the literature on the automated political event data collection and multilingual news monitoring has been considered. This Section provides a brief overview of the general workflow of several systems for political event data collection and coding that process few types of protest events, and of two early prototypes that focus on protest events. Our overview of multilingual news monitoring systems and approaches is given in [3].

A. Political Event Extraction Systems

Political event extraction systems aim at providing an unsupervised dynamically growing database of quantitative and qualitative features of different event types. In our work, we have been studying the mechanisms of the following projects: El:DIABLO[2], W-ICEWS[3], SPEED [7], and GDELT [6]. The access to SPEED and ICEWS data is currently limited. All of these systems but SPEED rely on the CAMEO ontology (Conflict and Mediation Event Observations Ontology [9]). This ontology encompasses events related to interstate political activity rather than on government-society interactions, and lists only 4 types of protest events. The Societal Stability Protocol of the SPEED project, in contrast, reflects many types of collective actions and government response. However, SPEED is not fully automatic, the fine-grained information representation and final decisions are supported by human experts.

[1] http://eventdata.psu.edu

[2] http://openeventdata.github.io/eldiablo/

[3] http://www.lockheedmartin.com/us/products/W-ICEWS/iData.html

The general workflow of the systems is given in Fig. 1:

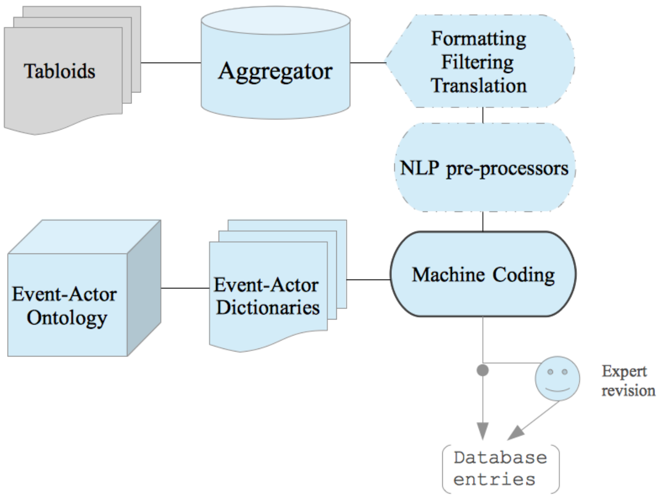


Fig. 1. General workflow of political event extraction systems

At the first step, news articles are gathered from online newspapers via a news aggregator, such as Factiva[4], Google, Nexis[5], Heritrix[6] or an own scraper. Most of the newspapers are currently presented in tabloid format. Tabloids are compact size, tend to emphasize sensational stories and cover a wide spectrum of political topics. Secondly, the resulting dataset undergoes the first-stage pre-processing, which includes duplicate articles removal, formatting, and translation, in case the articles have not been pretranslated by news providers. Shallow linguistic analysis, including tokenization, sentence segmentation, part-of-speech tagging, verb and noun phrase chunking, and named entity recognition, is performed using mostly open source packages, such as CoreNLP[7], a Java-based toolkit with a conditional random fields-based named entity recognizer, Apache openNLP[8], a maximum entropy and perceptron-based machine learning toolkit. ICEWS integrates Serif, a statistical entity and relation finder by BBN technologies[9] [1]. At the machine coding stage, events are detected on the basis of manually collected dictionaries of verb and noun phrases, connected to domain ontologies of actors and events. An Event of Interest (EOI) is a relation between two entities, the Source Actor and the Target Actor. Within a newspaper article, each sentence may include one or more events. The trigger concept is first detected (expressed by a verb or a noun phrase), and, secondly, the entities, expressed by noun phrases, on both sides of the relation, are extracted. Relation triples are acquired either via machine learning, or patterns (hand-crafted and bootstrapped). At the final stage, the extracted event data is stored in the database. In some systems (e.g., SPEED), the event data is subject to the analysis and refinement by human experts before storing.

B. *Protest Event Extraction Prototypes*

The latest attempts of computational sociologists to construct prototypes for protest event collection and coding are summarized as follows. [12] aggregate English articles (”The Guardian” archive) using a pre-defined keywords list via the Nexis engine, which yields poor results (only 68 out of 727 crawled articles address protest events), therefore, a postselection is needed. To this end, an active learning classifier is applied that uses as input the depency parsing output of the UIMA framework[10], and hidden topic modelling (HTM) performed with the mallet toolkit[11]. The reported joint performance of the postselection algorithm is 88.6% (Recall) and 70.1% (Precision). The event coding evaluation is not presented, the authors only mention some hints on the development of the algorithm. Together with the Stanford conditional random fields-based named entity classifier[12], they propose to identify the issue of the protest by measuring the distance between a given text and the corresponding hidden topic, and the protest form - using heuristic rules on the word space models output (dependency triples).

[5] uses two SVMs (Support Vector Machine classifiers), pre-trained on the DoCa (Dynamics of Collective Action dataset[13]), in order to (1) perform a binary classification of protest and non-protest articles, and (2) classify claims (protest reason), targets, dominant protest forms and initiating groups for each of the protests (multiclass classification). The third task consists in the classification of location, event size and organizations by training a named entity recognizer of the Stanford NLP group. No evaluation for the third task is presented. The reported binary classifier performance is 0.06% (Precision) and 0.86% (Recall). As for the second task, the highest F-measure value is 70% (target and claims identification), the lowest is under 50% (initiating group).

The summary of both approaches is presented in Tab.1:

TABLE I. PROTEST EVENT SELECTION & CODING SYSTEMS OVERVIEW

System	Wueest et al. (2013)	Hanna (2014)
Language Coverage	English	English
Training Set	”The Guardian”	”The New York Times”
Pre-processing toolkit	UIMA, HTM	–
Concept Hierarchy	–	DoCa
Postselection Algorithm	Active Learning	SVM (binary mode)
Coding Algorithm	NER & heuristics	SVM (multiclass)

The disadvantages of these systems can be roughly outlined as follows: (1) monolinguality; (2) news bias problem (limited coverage of news sources); (3) limited event descriptions; (4) need to construct large labeled datasets to feed the classifier.

The ability to process multilingual sources is a key functionality that ensures complementarity and a less biased view of events and opinions [11]. Also, as it has been proved by [8], cross-lingual information fusion improves the event extraction quality.

[4] http://new.dowjones.com/products/factiva/

[5] http://lexisnexis.com

[6] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

[7] https://github.com/stanfordnlp/CoreNLP

[8] http://opennlp.apache.org/

[9] http://www.raytheon.com/ourcompany/bbn/

[10] http://uima.apache.org

[11] http://mallet.cs.umass.edu/

[12] http://nlp.stanford.edu/software/CRF-NER.shtml

[13] http://web.stanford.edu/group/collectiveaction/cgi-bin/drupal/

Our pipeline uses generic linear patterns that rely on ontology-based multilingual gazetteers, thus, no training corpora are needed. It allows to extract data from various news sources in several languages (Bulgarian, French, Polish, Russian, Spanish, and Swedish), and code sentences for a variety of features that share their structural properties across languages. Together with the basic features, such as event type (mass demonstration, boycott, riot, etc..), location, event reason (the position of protesters towards an issue), date and actor (participants, initiators, targets), we plan to introduce text-based features for the size, duration, violence use, etc. to measure the event scale and importance of a given issue for the civil groups. Most political event extraction systems extract only four CAMEO protest events. Our ontology contains 31 entries for non-violent acts and 10 entries for violent attacks of civil groups.

III. Multilingual Protest Event Extraction Pipeline

The intuition behind the selection of headline as the analysis unit in this implimentation is that news titles tend to describe in a compact and clear fashion the key event concepts. According to the journalistic 5W rule, the answers to Who, What, Where, When and Why are usually given in the lead sentences of a news article [4]. [13] showed that in 80% of cases the noun in the title of a news article is the main argument of an event.

As the above systems, our pipeline includes two processing stages: event data selection and coding. Event selection implies crawling various multilingual sources, storage in JSON format, and primary pre-processing of text units. Event coding is the process of event data annotation and code assignment. In the present implementation, we do not apply any specific codes, only the ontology based annotation. The coding procedure includes two substages, namely: linguistic processing and output generation. Both tasks are accomplished within the open source GATE framework (General Architecture for Text Engineering)[14]. The output is made in the CSV format and ontology population with the labeled instances is accomplished in order to facilitate the processing of new data. The general structure is outlined in the Fig. 2.

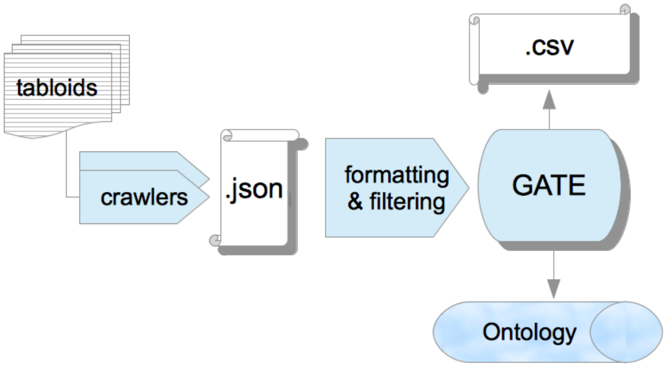


Fig. 2. General pipeline for multilingual protest event extraction

A. Event Selection

The crawlers for each of the languages have been developed within the Scrapy web crawling framework[15] in order to collect a multilingual corpus for further processing. They are able to extract news headlines and, optionally, the text body, date/time and source. An article is selected and stored in case of the mutual presence of several key phrases from two predefined lists in the headline. The first list includes protest TYPE names. The English equivalents are as follows: *demonstration, manifestation, protest, rally, action, boycott, strike, picketing, hunger strike, gathering, parade, procession, march, riot, revolt, civil disorder, civil unrest, rebellion, uprising, mutiny, insurgency*, and symbolic acts. The second list contains triggers for the co-occurring concepts, such as LOCATION (prepositions) and REASON (complex prepositions, such as ”in support of”, ”in defence of”, etc.). The primary pre-processing includes UTF-8 formatting and removal of total duplicate headlines, partial duplicate headlines and stopwords. Duplicate headlines are removed using NLTK (Levenshtein edit distance) and difflib Python packages.

B. Event Coding

Firstly, the multilingual headline set is tokenized and segmented. A multilingual ontology-based gazeteer lookup (Levenshtein edit distance based) and part-of-speech tagging are performed. Several taggers, such as Freeling, Russian Mystem, were integrated and tested until we finally opted for the TreeTagger[16] plugin for Bulgarian, French, Polish, Russian, Spanish, and The Stockholm tagger[17] for Swedish. The TreeTagger yields satisfactory results, however, the models obviously need domain-specific training. Special scripts have been written in order to adapt the output of the Stockholm tagger (and, earlier, the Freeling tagger) to serve as the input for the GATE pipeline.

The protest event ontology is constructed by a domain expert on the basis of the manual analysis of several thousand headlines. It covers the central protest event with its subtypes, attributes, and the subevents that often accompany it, such as protest antecedents (disruption warning, protest request, protest threat, time/route change), aftermath (economic damage, another event, authority response), etc.. We provide a detailed description of the ontology in [2].

The gazetteers are manually constructed and bootstrapped in a semi-automatic way using the Gazetteer List Collector[18]. The correctly extracted instances go directly to the ontology gazetteers in order to facilitate the further annotation. Also, a special-purpose JAPE[19] grammar has been built that can populate the main ontology with these instances, so that more

[14] http://gate.ac.uk

[15] http://scrapy.org

[16] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

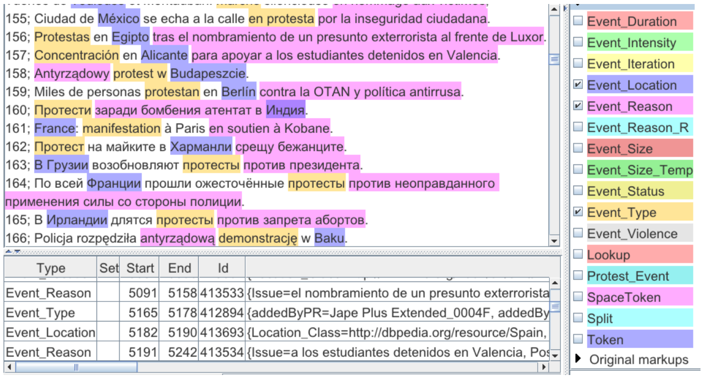
[17] http://www.ling.su.se/english/nlp/tools/stagger/stagger-the-stockholm-tagger-1.98986

[18] https://gate.ac.uk/releases/gate-5.1-beta1-build3397-ALL/doc/tao/splitch13.html

[19] https://gate.ac.uk/sale/tao/splitch8.html

advanced semantic annotation plugins (e.g., OntoRootGazeteer[20]) can be used. The output into the CSV format is generated using the configurable exporter.

The GATE GUI annotation sample is given in Fig. 3.



155; Ciudad de México se echa a la calle en protesta por la inseguridad ciudadana.
156; Protestas en Egipto tras el nombramiento de un presunto exterrorista al frente de Luxor.
157; Concentración en Alicante para apoyar a los estudiantes detenidos en Valencia.
158; Antyrządowy protest w Budapeszcie.
159; Miles de personas protestan en Berlín contra la OTAN y política antirrusa.
160; Протести заради бомбения атентат в Индия.
161; France: manifestation à Paris en soutien à Kobane.
162; Протест на майките в Харманли срещу бежанците.
163; В Грузии возобновляют протесты против президента.
164; По всей Франции прошли ожесточённые протесты против неоправданного применения силы со стороны полиции.
165; В Ирландии длятся протесты против запрета абортов.
166; Policja rozpędziła antyrządową demonstrację w Baku.

Type	Set	Start	End	Id	
Event_Reason		5091	5158	413533	{Issue=el nombramiento de un presunto exterrorista
Event_Type		5165	5178	412894	{addedByPR=Jape Plus Extended_0004F, addedBy
Event_Location		5182	5190	413693	{Location_Class=http://dbpedia.org/resource/Spain,
Event_Reason		5191	5242	413534	{Issue=a los estudiantes detenidos en Valencia, Pos

- □ Event_Duration
- □ Event_Intensity
- □ Event_Iteration
- ☑ Event_Location
- ☑ Event_Reason
- □ Event_Reason_R
- □ Event_Size
- □ Event_Size_Temp
- □ Event_Status
- ☑ Event_Type
- □ Event_Violence
- □ Lookup
- □ Protest_Event
- □ SpaceToken
- □ Split
- □ Token

Fig. 3. A sample annotation of a multilingual development dataset in GATE

Pattern-rule pairs are built in JAPE (Java-based pattern engine) formalism: cascaded finite-state transducers, where patterns are regular expressions over typed feature structures. The left-hand side (LHS) of a rule describes pattern constraints and the right-hand side (RHS) - annotation commands. Currently, we use Jape Plus Extended[21] that allows the use of additional constraints. Our multiphase grammar includes 12 phases. Patterns take into account the properties of noun phrase construction in each of the languages. First, the protest Event_Type is identified (verb, noun or adjective) and annotated with the corresponding ontology class using a single-pattern grammar. Next, the main descriptors are annotated: Event_Reason grammar (7 patterns), and Event_Location grammar (4 patterns). At the next stage, the rest of the concepts are highlighted in case they are present in a given sentence: Event_Size (5 patterns), Event_Duration (3 patterns), Event_Iteration (3 patterns), Violence_Involved (1 patterns), Event_Intensity (1 pattern), and Event_Status (1 pattern). Finally, the features of the mentioned annotations together with the Event_ID are added to the main Protest_Event annotation that covers the whole headline. An excerpt from the Event_Reason grammar code is as follows:

```
Rule: OntoLookup

({Event_Type})
({Token})[0,20]
({Lookup.majorType == Reason}):position
(CONTENT):issue
-->
:position{
 AnnotationSet pAS =
(AnnotationSet) bindings.get("position");
 AnnotationSet LookupAS = inputAS.get
("Lookup", pAS.firstNode().getOffset(),
pAS.lastNode().getOffset());
HashSet fNames = new HashSet();
    fNames.add("class");
    AnnotationSet ontoLookup =
LookupAS.get("Lookup", fNames);
...
```

IV. Evaluation

The evaluation of protest event selection and Event_Reason information block annotation has been performed by one multilingual domain expert. At this stage, we do not use multiple annotators' responses for two reasons: (1) in case of Event_Reason annotation evaluation, there will be a high inter-annotator agreement, because this slot has a clear natural language representation, and (2) current resources do not allow us to have several multilingual experts or unexperienced annotators for each of the languages.

A. Event Selection

For protest event selection task, we present the counts of the total number of messages, extracted by the crawlers per session, and the number of messages after the filtering of total headline duplicates and stopwords. As it can be seen in Tab. 2, the main portion is filtered out as total duplicates. Tab. 3 shows the number of manually checked instances per language after total duplicates filtering, the number of true negative reports, as well as the percentage of true positives.

TABLE II. The number of documents (headlines) in the language-specific datasets before and after the total duplicates filtering and stopwords removal

Language	Before Filtering	Total Duplicates	Stopwords
Bulgarian	4113 (528 kb)	1308	1306
French	8286 (615 kb)	1468	1242
Polish	5820 (591 kb)	1644	1561
Russian	7686 (673 kb)	4656	4654
Spanish	8678 (756 kb)	4683	4252
Swedish	3580 (180 kb)	705	695

TABLE III. The percentage of the reports that have been manually sorted out of the checked reports as unrelated to the topic.

Language	Checked	True Negatives	True Positives
Bulgarian	700	8	99 %
French	700	159	78 %
Polish	700	144	84 %
Russian	700	55	92 %
Spanish	700	93	87 %
Swedish	700	22	97 %

The number of true positives is lower for the French language, because the corresponding query contains more ambiguous substrings (”croisade”, ”marche”, ”blockage”, ”concentration”, ”contestation”, ”rassemblement”), than queries elaborated for other languages.

B. Event Annotation

Event annotation (Event_Reason slot) is evaluated using Precision and Recall metrics:

$$Precision = \frac{|G \bigcap C|}{|G|}, Recall = \frac{|G \bigcap C|}{|C|}$$

where G is the number of annotations extracted from all the documents (lead sentences) for a given information slot,

[20] https://gate.ac.uk/releases/gate-5.1-beta1-build3397-ALL/doc/tao/splitch13.html

[21] https://code.google.com/p/gateplugin-japeutils/wiki/JapePlusExtended

C is the amount of documents for which a given annotation corresponds to an expert annotation of the same slot.

The extraction evaluation on the Russian test set in terms of Precision, Recall, and F-measure is given in [2]. In this paper we present the results of an evaluation performed on the development set, which constitutes a collection of 400 multilingual headlines, manually selected from the main dataset for the purpose of parameter tuning. Each headline is information-rich and contains at least Event_Type, Event_Reason, and Event_Location blocks. A screenshot of this dataset is presented in Fig. 3. The evaluation results are shown in Tab. 4.

TABLE IV. EVENT_REASON EXTRACTION EVALUATION ON THE DEVELOPMENT SET.

Annotation Type	Precision	Recall
Event_Reason	0.96	0.98

V. CONCLUSION AND FUTURE WORK

The present paper summarizes the existing approaches to political and, specifically, protest data generation, and presents a pipeline for multilingual annotation of event features that share syntactic/semantic structure across European languages. It can be used independently or to enhance the runtime performance of expert human annotators. The pipeline is able to process around 500 sentences per 1 sec., which is faster than the pattern-based PETRARCH (150 sentences per 1 sec.). We use GATE annotation framework that has very well-known drawbacks (with the processing of complex grammars and large corpora), related to the runtime performance, however, patterns and gazetteers have simple structure that can be easily mapped to any other formalism and framework. As the future work, we plan to finally publish the evaluation of the whole set of features, connected to the ontology, and train event feature classifiers. In order to build topic clusters of reasons and capture the variety of event reason text representations, a classifier will be trained on the instances identified from the headlines and the underlying full reports. The articles will be clustered, according to their content similarity, and the produced reason clusters will be populated from the corpus, adding new reason text representations for a given topic.

REFERENCES

[1] Boschee, E., Weischedel, R., and Zamanian, A.: Automatic information extraction. In: Proceedings of the International Conference on Intelligence Analysis, McLean, VA, May 2-4 (2005)

[2] Danilova, V.: Ontology Building and Annotation of Destabilizing Events in News Feeds. In: Artif. Intell. Applications to Business and Engineering Domains (Monograph), ITHEA Publ., pp.137-146 (2014)

[3] Danilova, V., Alexandrov, M., Blanco, X.: A Survey of Multilingual Event Extraction from Text. In: Proc. of the 19th Intern. Conf. on Application of Natural Language to Inform. Systems (NLDB-2014), pp. 85-88 (2014)

[4] Danilova, V., Popova, S.: Socio-Political Event Extraction Using a Rule-Based Approach. In: Proc. of the 13th International Conference on Ontologies, DataBases and Applications of Semantics (ODBASE'2014), Springer, Volume 8842, pp. 537-546 (2014)

[5] Hanna, A.: Developing a System for the Automated Coding of Protest Event Data - Available at SSRN 2425232 (2014)

[6] Leetaru, K., Schrodt, P.: GDELT: Global Data on Events, Location, and Tone, 1979–2012. In: ISA Annual Convention, vol. 2 (2013)

[7] Nardulli, P., Althaus, S., and Hayes, M.: A Progressive Supervised-learning Approach to Generating Rich Civil Strife Data (Forthcoming in Sociological Methodology)

[8] Piskorski, J., Tanev, H., Atkinson, M., van der Goot, E., Zavarella, V.: Online News Event Extraction for Global Crisis Surveillance. In: N.T. Nguyen (Ed.): Transactions on CCI V, LNCS 6910, pp. 182-212 (2011)

[9] Schrodt, P.: "CAMEO: Conflict and Mediation Event Observations Event and Actor Codebook." Event Data Project, Department of Political Science, Pennsylvania State University (2012)

[10] Schrodt, P., Yonamine, J.: Automated Coding of Very Large Scale Political Event Data. In: Armed Conflict Location & Event Data Project (ACLED), Guide to Dataset Use for Humanitarian and Development Practitioners, pp. 3-9 (2015)

[11] Steinberger, R.: A survey of methods to ease the development of highly multilingual text mining applications. In: Language Resources & Evaluation 46: 155-176 (2012)

[12] Wueest, B., Rothenhäusler, K., Hutter, S.: Using Computational Linguistics to Enhance Protest Event Analysis - Available at SSRN 2286769 (2013)

[13] Wunderwald, M.: Event Extraction from News Articles (Diploma Thesis), Dresden University of Technology. Dept. of Computer Science (2011)