Knowledge-based highly-specialized terrorist event extraction

Jakub Dutkiewicz, Czesław Jędrzejek, Jolanta Cybulka, Maciej Falkowski

Institute of Control and Information Engineering,

Poznan University of Technology,

Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland

{firstname.lastname}@put.poznan.pl

Abstract. In this paper we present a prototype of a system aimed at event extraction using linguistic patterns with semantic classes. The process is aided with an auxiliary tool for mapping verb statistics across messages. The sentence analyzer uses linguistic associations, based on VerbNet across the message and between messages' sentences to select semantic role fillers. We restrict ourselves to the coverage of one event type only – namely a kidnapping – and to two events template slots (semantic roles): a perpetrator and a person_target (a human target). We designed rules involving semantic role filling using previous works on coreference. We used the Sundance parser and AutoSlog extraction patterns generator. Then we applied the semantic role filler and event resolution tool SRL Master. Our approach yields high performance on the MUC-4 data set.

Keywords. knowledge-based information extraction, semantic roles, terrorist event discovery

1 Introduction

Event extraction is one of the most important tasks of knowledge discovery. It may be regarded as the core of knowledge-based systems that aim at providing the public (people, organizations, government agenda etc.) with condensed and filtered information concerning events. These events are described in texts written in natural language, thus the posted problem is related to the issue of information extraction (IE). Particularly, the task is to extract data concerning the described action (the event) and its arguments (called event roles). To implement the considered task different approaches are applied. They can be classified according to the provenance of the approach (pattern-based linguistic ones vs. classifier-based (statistical) methods) or to the 'openness' of it (fully open extraction vs. trained with the use of corpora one). The next important classification criterion is the nature of the context of the extraction, namely locality (one sentence only) or a larger context that takes into account consecutive sentences (a discourse). In many cases the hybrid methods are used that combine the different approaches. The open extraction systems (operating across one sentence context) scale well to the open corpora [1,3], especially that acquitted from the Web. But the most accurate IE systems are domain-specific, that use linguistic patterns and are somewhat trained with the aid of statistics. Our work follows the

latter approach in that we use a training domain-specific corpus. Let us characterize it briefly.

Due to a series of DARPA Message Understanding Conferences (MUCs), significant progress in pattern-based (NLP based) extraction technologies has been achieved. In this work we capitalise on the results of MUC-3 and MUC-4 ([10] that were held in 1991-1992) conferences, which used news reports corpus (MUC corpus) on terrorist activities in Latin America. MUC Conferences developed standards for evaluation, e.g. the adoption of metrics like precision and recall.

The goal of MUC was to extract from texts an information concerning 7 classes of terrorist events: Attack, Kidnapping, Hijacking, Bombing, Arson, Robbery and Forced Work Stoppage, plus several variations on each (for accomplished, threatened and attempted incidents). The process of extraction was augmented by the knowledge frames (event templates) generation. Every such template consisted of 24 attributes-slots. A document (a multi-sentenced message concerning an event) could be labeled with more than one template type. The MUC-4 corpus consists of 1700 documents, from which 1300 (DEV) were used in MUC-4 for training, 200 documents (TST1+TST2) were used as a tuning set, and the last 200 documents (TST3+TST4) were applied as the test set. The resulting knowledge base frames are called "key templates". We filter out messages concerning one event type only, namely the kidnapping. Also, from among 24 slots we consider the two of them: a perpetrator and a person_target.

The main contributions of the presented paper are:

- a method of comparing events to check whether a given two events are in fact identical or whether they are different, on the basis of semantic typing (semantic classes) of event's arguments; it relies on using several types of rules, namely atomic, filling thematic role rules and whole events comparing rules; the method may be also used in coreference resolution
- an implementation of a corpus crawling tool that looks for words/phrases that lexicalize the kidnapping event
- additional lexical rules related to identification of victims and perpetrators.

The paper is organised as follows. Section 2 contains some notes concerning related works. In Section 3 our extraction method is presented. Section 4 describes a prototype implementation of the Word-statistics tool and its use. Section 5 demonstrates our information extraction results. In section 6 we give the concluding remarks and mention on our future work.

2 Related works

The main drawback of open information extraction [3] is that it uses the natural language features which do not classify (semantically type) arguments of an extracted relation. Additionally, in such methods the syntactic patterns (for example, regular expressions) do not match verb arguments that are distant from the verb phrase in a sentence. These are the features having the great negative impact on the ability to compare events (whether they are identical or not) described in the different sentences. In our work we avoid this drawback.

Authors of [4] use the language resources (dictionaries) to obtain sets of words that are relevant to the semantic class (a type of a verb argument). Having such extensionally defined types (semantic classes) they use them in the extraction process. In this work it is also shown how to apply such classes in the process of events comparison.

The method of event's comparison is also described in [5]. Here, the authors compare them (and extract their arguments) on the basis of head parts of noun phrases. For example, the events described in the following two sentences:

1) A customer in the store was shot by masked men.
2) The two men used 9mm semi-automatic pistols.

are in fact the same due to the fact that they use the same word "men". In our approach the events may be unified (or differentiated) on the basis of the membership (non-membership) of two used ("linking") words to the same semantic class. Also, it is not known, which pairs of sentences should be analyzed according to the event (we describe this problem later on).

3 The extraction method

3.1 Preliminary definitions

At first, let us give some definitions of the terms used in the paper. They are as follows.

Event (denoted by E_n, where *n* stands for event's name) is an entity representing the event (conceptually it is an *occurrent* that plays the central role in some situation, which represents a state of affairs) described in the text. The event is connected with a syntactic phrase (a verb phrase) that helps to identify it in a sentence, which is called an anchor. Also, there are some participants in the event – we identify them via thematic roles that are arguments of an anchoring phrase.

Anchor (marked as A_k, where *k* stands for an anchor name) is a verb or a verb phrase, which appearance in a derivation (i.e. a syntactically parsed sentence) triggers the process of recognition of an event (such as, for example, the kidnapping).

Thematic role (a semantic role label, marked as R_m, where *m* is a role name) is an entity representing an argument of a verb or a verb phrase (an anchor) denoting the event. For example, there may be such roles as Agent (in our considerations, a perpetrator), Patient (a victim), Instrument, Location, Time and others.

Role filler is a text phrase that instantiates a thematic role in the text (marked with the symbol R_pF_v, where *p* is a role name and *v* identifies a filler).

Syntactic similarity. Let us assume that the two argument function of syntactic similarity simsyn (W_1, W_2), while given two words (or phrases) as arguments returns a binary value *true* or *false*. The function will return the *true* value if W_1 and W_2 have the same syntactic properties (i.e. number and gender), otherwise it returns *false*.

Semantic class (denoted by C_s, where *s* is a class name) is defined as an entity that is expressed by all of its verbalizations. For example, the verbalizations of the semantic class concerning kidnapping are $C_{kidnapping}$={kidnap, seize, abduct, capture, intercept, take hostage}. It should be noted that we do not use all the meanings of the listed words, but only these fitting to a specific context.

Atomic formula is a triple of the form <sub, pred, obj>, where *sub* means the subject of the sentence (and semantically it may play a thematic role R_m), *pred* means the predicate (represents an event in terms of a certain semantic class C_s) and *obj* means the object (semantically playing a role R_p). An atomic formula could be considered as a rule representing a fact.

Let us illustrate the introduced notions with the exemplary message from DEV-MUC3-0018 (the text in this corpus is given in an upper case). We decorated the text with roles, role fillers, events and anchors. One of the considered sentences is:

OQUELI, LEADER OF THE NATIONAL REVOLUTIONARY MOVEMENT (MNR) AND HILDA FLORES, A GUATEMALAN SOCIAL DEMOCRATIC LEADER($R_{victim}F_1$) WERE ABDUCTED($E_{kidnapping}A_{kidnapping1}$) AND KILLED IN JANUARY($R_{time}F_1$) BY UNIDENTIFIED INDIVIDUALS($R_{perpetrator}F_1$) IN GUATEMALA CITY($R_{location}F_1$) AS THEY WERE HEADING TO THE LA AURORA AIRPORT.

Assume that there exists another sentence concerning the same event but with the new fillers for the victim and perpetrator roles:

IT TURNED OUT THAT POLITICIANS($R_{victim}F_2$) WERE KIDNAPPED($E_{kidnapping}A_{kidnapping2}$) BY URBAN TERRORISTS OF FARABUNDO MARTI NATIONAL LIBERATION FRONT($R_{perpetrator}F_2$).

After decorating the two sentences we are to check, whether two pairs: $E_{kidnapping}A_{kidnapping1}$ and $E_{kidnapping}A_{kidnapping2}$ concern the same event. We will show how to approach this issue in section 2.3.

We are motivated by VerbNet (VN) [1] thematic/semantic role methodology. VerbNet verb classes are organized according to the syntactic behavior of verbs. VerbNet uses 109 verb classes and 29 semantic role labels for arguments of the <sub, pred, obj> triple pattern (which resembles our atomic formulae). We adhere to VerbNet semantics rather than to ontologies, because we are not aware of any publicly available ontology with adequate expressive power and rich verbalization of classes (ontological entities). We are in the process of using our CATIE ontology for the general extraction of facts from MUC-4 corpus [6].

We are interested in such event specifying verbs as: kidnap, abduct, seize (VN sense no 3), take (by force) (VN sense no 21, http://verbs.colorado.edu/verb-index/vn/steal-10.5.php#steal-10.5; sense number 3: take or capture by force or authority) belonging to class steal-10.5. However, instead of a role Agent [+animate | +organization] we need a role Agent/Patient [+person | +a group of persons | +organization]. In Unified Verb Index collection (VerbNet generalization) the word capture belonging to class steal-10.5.1 (http://verbs.colorado.edu/verb-index/wn/wordnet.cgi?v3-0.capture.1.capture-2:36:00#1) apparently has not been assigned a meaning kidnap.

3.2 Basic rules for identifying thematic roles

The next type of rules (besides the earlier described atomic formulae that represent facts) says that as the direct anchors we use all the interesting verbs ($C_{kidnapping}$) in the past tense forms. Using a special function that retrieves a predicate of a given triple, namely predicate_of(<s,p,o>) = p, we denote such rules as triples of the form: <predicate_of(<s,p,o>), tense_of, "Past">. We assume that tense_of is a built-in predicate

representing verb tenses, i.e. “Past” and “Past Participle”. Another built-in predicate, named voice_of, represents voice of a verb phrase, namely “active_voice” and “passive_voice”. The third built-in predicate, named plays, represents a fact concerning the deduced thematic role of a subject and an object of some triple (as it was assumed we only consider the agentive role (a perpetrator) and the patientive (beneficiary) role – a victim).

Now we are ready to give the rules to identify thematic roles of a predicate given in the past tense form. We are concerned with predicates expressed by verbs being members of a $C_{kidnapping}$ semantic class.

The first rule states that for a given triple if its predicate is in the past tense and in the active voice then the subject plays the agentive thematic role of a perpetrator while the object plays the patientive thematic role of a victim (a kind of a person_target). The rule (1) is as follows:

<predicate_of(<sub,pred,obj>), tense_of, “Past”> ∧
<predicate_of(<sub,pred,obj>), voice_of, “active_voice”> ⇒
<sub, plays, “agentive_role”>∧ <obj, plays, “beneficiary_role”> (1)

The second (2) rule differs in the voice specification only that influences the order of the atomic formulae in the conclusion. The rule is as follows:

<predicate_of(<sub,pred,obj>), tense_of, “Past”> ∧
<predicate_of(<sub,pred,obj>), voice_of, “passive_voice”> ⇒
<sub, plays, “ beneficiary_role”> ∧ <obj, plays, “ agentive_role”>. (2)

3.3 Rules for event identification

In many cases information about certain roles and events is included in several sentences. Thus, matching different phrases to one thematic role constitutes one of a key tasks. We define a set of rules to identify such cases and eventually we either unify different events or differentiate them (the are_different predicate). One of these rules bases on two sentences with a verb phrases denoted as two pairs containing an event and an anchor, $E_{n1}A_{m1}$, $E_{n2}A_{m2}$. Each of these sentences contains a phrase that represents a filler of the same role, namely $R_{p1}F_{k1}$, $R_{p1}F_{k2}$. To activate such a rule we need to find at least two sentences with these role fillers and event anchors. If we happen to find more than two sentences of such a kind, we need to analyze them in pairs. To describe such a rule, we need to define two predicates. The “belongs_to” predicate is used if a given phrase belongs to a certain semantic class (this means that the main word in the phrase is a member of the considered class). The “is_equal_to” predicate decides whether either two semantic classes contain the same set of elements or role fillers are syntactically equivalent.

The process of analysis starts with searching of described pair of sentences. Let us denote the anchor and the role filler that were found in the first sentence as R_1F_1 and E_1A_1, and the anchor and the role filler found in the second sentence as R_1F_2 and E_2A_1. Once we have found these pairs we need to decide whether the described event anchors belong to the same semantic class (denoted as C_1). This is formalized as:

$<E_1A_1$, belongs_to, $C_1> \wedge <E_2A_1$, belongs_to, $C_1>$.

This basic condition should be considered as preemptive and its result decides if we are going to consider a pair of sentences as worth of executing this rule on.

The second part of the analysis starts with determining if role fillers belong to classes that are different, but there exists some relation between those classes. Furthermore we need to check if role fillers have the same syntactic properties. If those conditions are true, we can assume that phrases describe the same event. Additionally, there exists some relation among semantic classes, which may also be projected on role fillers (in particular it may be a subsumption). Let us formalize these considerations in the form of rule (3). In this rule, we mark "some relation" as a variable "?rel".

$$<R_1F_1, \text{belongs\_to}, C_2> \wedge <R_1F_2, \text{belongs\_to}, C_3> \wedge <C_2, ?\text{rel}, C_3> \wedge$$
$$\neg <C_2, \text{is\_equal\_to}, C_3> \wedge \text{simsyn}(R_1F_1, R_1F_2)$$
$$\Rightarrow$$
$$\text{are\_the\_same}(E_1, E_2) \wedge (R_1F_1, ?\text{rel}, R_2F_2) \quad (3)$$

However, if role fillers belong to the same class, but are different or role fillers have different syntactic properties, it is necessary to classify two events as different (4):

$$(<R_1F_1, \text{belongs\_to}, C_4> \wedge <R_1F2, \text{belongs\_to}, C_4> \wedge \neg <R_1F_1, \text{is\_equal\_to}, R_1F_2>)$$
$$\vee \neg \text{simsyn}(R_1F_1, R_1F_2)) \Rightarrow$$
$$\text{are\_different}(E_1, E_2). \quad (4)$$

We illustrate that rule with the following examples.

Example 1

There are two consecutive sentences in the message:

1) John Smith ($R_{victim}F_1$) has been kidnapped ($E_{kidnapping1}A_1$).

2) President ($R_{victim}F_2$) was taken hostage ($E_{kidnapping2}A_2$) by unknown perpetrators.

The preemptive constraints are:

<"kidnap", belongs_to, $C_{kidnapping}$> $\wedge$ <"take_hostage", belongs_to, $C_{kidnapping}$>.

The following rule activation captures lexical associations between two neighboring sentences by pairing as similar each noun in the role of a victim (person_target). This is similar to lexical bridge features used in [5]. The rule for those sentences goes as following:

<"John Smith", belongs_to, C_{Person}> $\wedge$ <"President", belongs_to, $C_{Politician}$> $\wedge$
<C_{Person}, represents, $C_{Politician}$> $\wedge \neg$ <"John Smith", is_equal_to, "President"> $\wedge$
simsyn("President", "John Smith")
$\Rightarrow$
are_the_same($E_{kidnapping1}$,$E_{kidnapping2}$).

As the result we obtain a fact (an atomic formula) of the form:
<"John Smith", represents, "President">.

The confidence of this rule could be measured in distance between the considered sentences (thus the distance is measured in the number of sentences). In particular this rule may be used only to analyze consecutive sentences.

Example 2

We have three sentences, not necessarily in one document.

1. Ricardo Alfonso Castellar, mayor of Achi,($R_{victim}F_1$)who was kidnapped($E_{kidnapping1}A_1$) on 5 January, apparently by Army Of National Liberation guerillas, was found dead.
2. Castellar($R_{victim}F_2$)was kidnapped($E_{kidnapping2}A_1$) by a group of armed men.
3. A politician condemned kidnapping($E_{kidnapping3}A_1$) of mayor of Achi($R_{victim}F_3$).

In this case we need to process sentences in pairs. First, we take sentences 1 and 2. We execute the rule and as a result we get the unification of $E_{kidnapping1}$ and $E_{kidnapping2}$. This means that unification of $E_{kidnapping3}$ event, with both of the previous events would be redundant and we just need to clarify if $E_{kidnapping3}$ could be unified with any of those events. However, if $E_{kidnapping1}$and $E_{kidnapping2}$ would not be unified, all events need to be compared separately. In this case we get three fillers of the victim role, furthermore the relation between those fillers is quite specific. That relation could be marked as "is_substring_of". The left-hand side argument of this relation is always less expressive then its right-hand side and thus we could find the most expressive filler – "Ricardo Alfonso Castellar, mayor of Achi".

Our method of unification is conceptually more powerful than the so far used for coreference resolution (for example in [11, 9]). But so far it is used only for establishing the agreement of semantic classes and also the noun-pronoun agreement features, that means features 2-3 and 8 out of 12 features proposed in [11].

3.4 Additional lexical rules

The examples shown in the previous subsection illustrate the need for rules that go beyond search of sentences with verb phrases corresponding to event related semantic class. To make the task of identifying event easier for the annotators, it is necessary to use the secondary semantic class containing words that are in a fuzzy relation to the core event term. We introduce a class:

$C_{fuzzy\_kidnapping}$ = {disappear, release}

Following the Automatic Content Extraction (ACE) Programme guidelines:
An **event trigger** refers to the term within the event mention that most clearly expresses the occurrence of the event instance and is based on direct anchor – corresponds to $C_{kidnapping}$.
An **event mention** refers to the sentence within which an event instance is reported – corresponds to $C_{fuzzy\_kidnapping}$. An event can have multiple mentions associated with it. Apart from the sentence that initially reports the event, other coreferring sentences that contain anaphors of events (such as pronouns and definite descriptions of previously mentioned events) are taggable mentions of that event [9].

In general there always exists a direct connection between roles of events corresponding to C_1 and $C_{fuzzy\_1.}$ For example a victim of kidnapping directly corresponds to a subject of releasement or disappearance. To measure the confidence of fuzzy classes we look at the statistics of all words/stems in various part-of-speech forms, which directly or indirectly could indicate an event of kidnapping. They are words

corresponding to $C_{kidnapping}$ and $C_{fuzzy\_kidnapping}$ classes – verbs for kidnap (heads of verb phrases) in the past tense or attributive kidnapped, verbs in the past tense, verbs (infinitve, -ing form for a verb, gerund), nouns related to an act of kidnapping or a perpetrator, namely:

kidnap, kidnapping, kidnapped, kidnapper
stem *seiz, seized, seizing,*
abduct, abducted, abducting,
stem *captur, capturing, captured,*
intercept, intercepting, intercepted,
stem *releas, released, releasing,*
disappear, disappeared, disappearing,
take/hold hostage.

Finally, we apply coreference rules for both $C_{fuzzy\_kidnapping}$ and $C_{kidnapping}$ semantic classes.

Example 3:

1. Ricardo Alfonso Castellar($R_{victim}F_1$), mayor of Achi, was released(E_1A_1) on 15 January.
2. Kidnapping(E_2A_1) of Castellar($R_{object}F_1$) was a brutal act.

Even though events E_1 and E_2 belong to different semantic classes we can unify specific role fillers within those events.

4 Word Statistics Tool

The process of designing pattern-based linguistic rules is a very tedious work, what constitutes the main disadvantage of such methods. To alleviate a burden we implemented a MUC Word Statistics Analyzer (Figure 1). The tool realizes several useful functions:

1) it presents graphically statistics of words across a document or a corpus
2) and it displays in two separate panels fragments of text pertaining to this statistics.

The considered in the paper extraction method relies on the quality of verb argument's typing (semantic classes). To obtain good results concerning the extensions of semantic classes $C_{kidnapping}$ and $C_{fuzzy\_kidnapping}$ we designed and implemented a statistic tool. It estimates the frequency of words (exactly, their stems) occurrences in the message or in the whole corpus. The tool also enables the analysis of sentences (or message) across which the stems appear. In the upper right corner of the screen given in Figure1 the histogram is located that depicts the number of a word (stem) occurrences in the message and in the sentence. The exemplary message is shown in the lower left corner. In the bottom panel the list of sentences is located in which the stems with

different endings appear, for example: a stem kidnap, end words kidnapped, kidnapper or kidnapping.

Summing up, by the quick inspection of the frequency of appearance of words and their correlation and varying the trigger term lists we can assess effectiveness of linguistic features.

5 Results

There are five overall IE related tasks that evolved from MUC.

- Named entity (NE) aims to extract all instances of persons, organisations, locations, dates, times, percentages and monetary entities.
- Coreference (CO) given a set of entities, this task aims to generate a set of entity coreference chains, such that mentions that coreference to the same entity appears in the same chain.
- Template element (TE) aims to extract all entity attributes. As an example, for the entity mention \ Castellar ", the aim is to extract its name (\Ricardo Alfonso Castellar, "), type (\PERSON") and descriptor (\the mayor of Achi").
- Template relation (TR) aims to extract all well-defined facts from each newswire text. In MUC-4 this was related to the knowledge frame (24 slots) of 8 terrorist type of events. In MUC-7 the facts were limited to relationships with organisations: employee of, product of and location of.
- Scenario template (ST) aims to extract pre-specified event information from anywhere in the given text, and relate it to the particular organisation and person entities etc. involved in the event.

The figures presented in this table are based on the performance levels of systems participating in the MUC evaluations. More detailed figures can be found in Table 1.

Table 1. MUC evaluation tasks

Year	Evaluation	MUC Tasks				
		NE	CO	TE	TR	ST
1991	MUC-3					F< 58%
1992	MUC-4					F<56% [9]
1995	MUC-7	F< 94%	F< 62%	F< 87%	F <76%	F< 51%

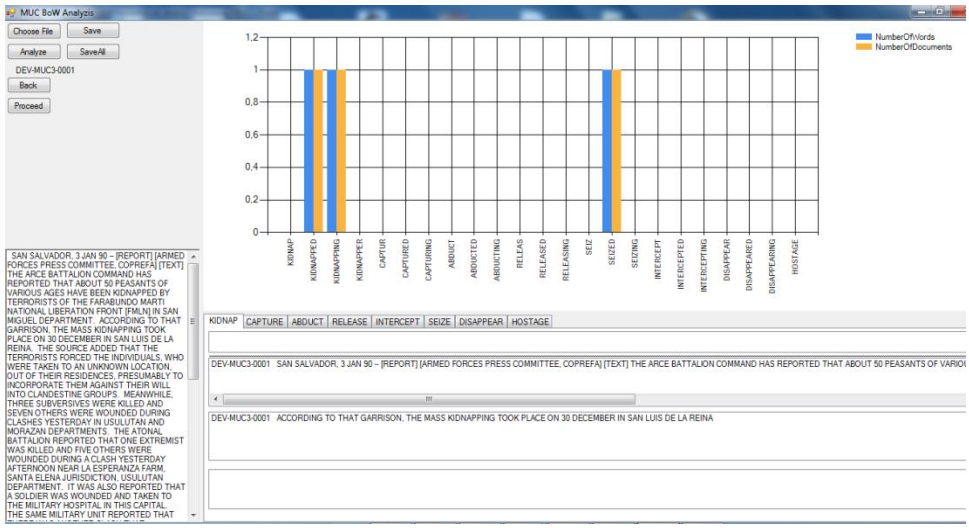

Figure 1. A snapshot of the results of the MUC Word Statistics Analyzer.

For many years these results have not been significantly improved. Only recently a significant progress [9,11] has been made.

There appear 159 events resolved as kidnappings out of 1700 documents as a result of assessment of the MUC-4 community [7].

We define the following numbers or word occurrences:

X1: at least a single occurrence of words from $C_{kidnapping}$ or $C_{fuzzy\_kidnapping}$

X2: only from $C_{kidnapping}$ at least once

X3: only from $C_{fuzzy\_kidnapping}$ at least once

X4: from $C_{kidnapping}$ at least once and from $C_{fuzzy\_kidnapping}$ at least once together

X5: only from $C_{kidnapping}$ ending with –ed at least once

X6: only from $C_{fuzzy\_kidnapping}$ ending with –ed at least once

X7: as in X1 from $C_{kidnapping}$ at least once and from $C_{fuzzy\_kidnapping}$ at least once, together ending with –ed

X8: only from {kidnap} set

X9: only kidnapped

Y1- Y9: occurrence as for X but for the set of documents that do not belong to a kidnapping event.

Table 2. Statistics of MUC evaluation tasks

	Number of occurrences								
Documents concerning the kidnapping	X1	X2	X3	X4	X5	X6	X7	X8	X9
	144	44	10	47	48	13	31	81	65
Documents not concerning the kidnapping	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9
	394	212	128	54	156	108	30	83	32

The detailed analysis of these results will be presented at the Challenge event. We used the Sundance and AutoSlog systems for syntactic parsing and extraction patterns generation [12] together with Name Entity Extraction (with slightly modified dictionaries). Then we applied the semantic role filler and event resolution tool SRL Master.

Table 3: The most effective patterns as determined from DEV 1-1300 texts

NP = EN	**VP(S)**	**PP(BY) = Perp**
NP = EN	**VP(S)**	**PP(IN) = Location**
NP = EN	**VP(S)**	**NP(Date) = Date**
NP = EN	**PP(OF) = Victim**	
NP = EN	**NP(Date) = Date**	
NP = EN	**PP(IN) = Location**	
NP = Victim	**PVP**	**PP(ON) Date**
NP = Victim	**PVP**	**PP(ON) Location**
NP = Victim	**PVP**	**PP(BY) = Perp**
NP = Victim	**PVP**	
NP = Victim	**PVP**	**NP(Date) = Date**
NP = Victim	**PVP**	**PP(IN) Location**
NP = Victim	**PVP**	**NP(Loc) = Location**
Pron = Victim	**PVP**	**PP(BY) = Perp**
Pron = Victim	**PVP**	**PP(IN) = Location**
Pron = Victim	**PVP**	**NP(Date) = Date**
Pron = Victim	**PVP**	
Pron = Victim	**PVP**	**PP(ON) = Date**
NP = Perp	**VP ActInf**	**NP= Victim**
NP = Perp	**VP**	**NP=Victim**
NP = Perp	**VP**	**PP(IN) = Location**
NP = Perp	**VP**	**PP(OF)= Victim**
NP= Victim	**AdjP**	**NP(Date) = Date**
Pron	**PVP AuxVP**	**NP = Victim**
NP= Victim	**AdjP**	**PP(BY) = Perp**

In Table 3 the meaning of symbols is the following: EN= event name (e.g. kidnapping, crime, etc.) – there are anchors, in all other patterns VP are anchors, NP = noun phrase, VP = verb phrase, PVP = passive verb phrase, AdjP=adjective phrase, PP= prepositional phrase starting with specific prepositions, Pron= noun phrase represented by a pronoun, Perp=perpetrator.

Effectiveness of our system is due to several factors:

- Our patters are mostly triples, whether most previous works were based on syntax patterns consisting of 2 elements, see *e.g* Fig. 1 of [13].
- Non-triple patterns are more likely to generate extraction of nonrelevant patterns. For a pattern to be relevant we need to have at least either of two: location, date sentence part (first sought in a simple sentence, then in the complex sentence, and finally in adjacent sentences.
- One of the main contributions of this work is the introduction of VP(S) = supplementary verb phrase (particularly effective involving NP=EN are: *take place*, *claim responsibility*, *be responsible for*, *carry out*. To a lesser degree this helps to identify perpetrators and victims.

The correctness of extraction in this paper is providing all of the following kidnapping event roles (recall): *perpetrator individuals*, *perpetrator organizations*, *human_target/victim*, *location* and *date*. These roles are narrower than 24 slots of the MUC-4 contest.

Table 4 presents the recall for the kidnapping events (here the same events in different documents are counted separately, similarly as for MUC-4 evaluation).

Table 4: Recall for the kidnapping events for the MUC-4 development and test sets

Recall Measure [per cent]	
DEV set	TST sets
78	73

The recall numbers are significantly higher than in the MUC-4 contest (where the best contribution achieved around 60% for both precision and recall), but achieved for the easier task and for only one type of a terrorism event. They are also higher than in Table 3 of [5].

The system is presented at http://draco.kari.put.poznan.pl/ruleml2013_Extraction.

6 Conclusions

The recent wave of methods [11,9,8,3,4] is capable of significant improvement of extraction measures. The MUC Conferences provided benchmarks that decrease arbitrariness of a given method evaluation. For example open extraction system ReVerb gives a good precision but a poor recall [3]. We plan to apply against the full MUC-4 benchmark. The MUC Word Statistics Analyzer would be helpful for this task. There

are improvement possibilities in using the probable better syntax parser, Named Entity Recognition and using a wider set of coreference comparison.

Our choice of anchor words can be more optimal. In general, our patterns presented in Table 3 are more compatible with ontology-driven extraction than purely linguistic methods. Rather than use one general dictionary as used by most MUC related works, we can have lexicalization specific to ontology element. We are working in this direction.

Acknowledgement. This work was supported by the Polish National Centre for Research and Development (NCBR) No O ROB 0025 01 and DS 45-085/13 and DS-PB grants. We would like to thank Prof. Ellen Riloff for making Sundance and AutoSlog tools available to us, and Bartosz Zaremba for calculating some statistics.

References

1. Bonial, C., Corvey, W., Palmer, M., Petukhova, V., and Bunt, H. A Hierarchical Unification of LIRICS and VerbNet Semantic Roles. Proceedings of the ICSC Workshop on Semantic Annotation for Computational Linguistic Resources (SACL-ICSC 2011), Sep, 2011.
2. Etzioni O., Banko M., Soderland S., and Weld D. S. 2008. Open information extraction from the web. *Commun. ACM* 51, 12 (December 2008), 68-74.
3. Etzioni O., Fader A., Christensen J., Soderland S., and Mausam: Open Information Extraction: The Second Generation. IJCAI 2011:3-10.
4. Huang, R. and Riloff, E.: Multi-faceted Event Recognition with Bootstrapped Dictionaries, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013).
5. Huang, R. and Riloff, E.: Modeling Textual Cohesion for Event Extraction, Proceedings of the 26th Conference on Artificial Intelligence (AAAI 2012).
6. Jedrzejek C., Cybulka J., CATIE ontology for the MUC-4 events extraction, in progress.
7. Lehnert, W.G., Cardie, C., Fisher, D., McCarthy, J., Riloff, E. and Soderland, S., Evaluating Information Extraction System, submitted to (Journal of Integrated Computer-Aided Engineering), 1(6), (1995), pp. 453-472.
8. Nakashole N., Weikum G., Suchanek F. M.: PATTY: A Taxonomy of Relational Patterns with Semantic Types. EMNLP-CoNLL 2012: 1135-1145.
9. Naughton M. Sentence-Level Event Detection and Coreference Resolution. School of Computer Science and Informatics, University College Dublin, PhD Thesis: October 2009.
10. Proceedings of the 4th Conference on Message Understanding, MUC 1992, McLean, Virginia, USA, June 16-18, 1992.
11. Soon, W. M., Ng H. T., and Lim D. C. Y. (2001). Learning approach to coreference resolution of noun phrases. Computational Linguistics 27 (4), 521–544.
12. Riloff E., Phillips M.. 2004. An Introduction to the Sundance and AutoSlog Systems Technical Report UUCS-04-015, School of Computing, University of Utah, http://www.cs.utah.edu/~riloff/pdfs/official-sundance-tr.pdf.
13. Patwardhan, S. and Riloff, E. (2006) "Learning Domain-Specific Information Extraction Patterns from the Web", ACL 2006 Workshop on Information Extraction Beyond the Document.