Combining Different NLP Methods for HUMINT Report Analysis

Constantin JENGE, Silverius KAWALETZ, Ulrich SCHADE
Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE)
Neuenahrer Str. 20
53343 Wachtberg-Werthhoven
Germany

{constantin.jenge|silverius.kawaletz|ulrich.schade}@fkie.fraunhofer.de

ABSTRACT

In this paper we present a combined approach to the automatic (pre-) analysis of intelligence reports. The combination encompasses information extraction (IE) and information enrichment by means of ontologies. The combined approach proves to yield superior results compared to standalone IE. For our work we mainly use open standards and open source software. For the purpose of IE, for instance, we use the GATE system, whereas our ontology work is based on the W3C OWL standard and the Protégé ontology editor.

1. MOTIVATION

In today's deployments military decision makers at all echelons have to cope with an unprecedented volume of information. Technological progress accounts for an increase in available information to a degree that no human can master. This observation is certainly true for data from sensors, for data gathered by tapping into diverse communication channels (SIGINT) as well as for the ever increasing stream of HUMINT information. While processing and analysing the masses of SIGINT data poses a challenge of its own, we assume in the context of this paper that vital pieces of information will at some point be transformed into a natural language representation. For example, sensor data is most often meaningful to only a handful of highly specialized personnel, who render their findings in some form of natural language. Therefore we focus on the automatic analysis of intelligence data in the form of natural language text. The result of this automatic analysis is represented in a form that helps the user to find those pieces of information that she needs to know.

2. PRELIMINARY WORK

Our process of (pre-) analyzing natural language reports starts with information extraction (IE) [5] based on the work of Hecking who applied IE techniques to the analysis of battlefield and HUMINT reports [9]. For information extraction, we use the freely available open-source tool GATE [2, 7], where we run our data through the standard IE processing pipeline. This pipeline consists of the following elements:

1. A tokenizer that determines individual tokens of the text, i.e. single words, numbers, abbreviations and punctuation marks.

2. A gazetteer that compares the tokens to elements of several lists which contain names of various types. There are usually lists for person names, organisations, countries, places, villages and the like. Tokens matching one or more elements in the list will be annotated with the respective type, e.g. *female forename*.

3. The sentence splitter determines the boundaries of sentences, which is less trivial than it may seem at first glance. A certain built-in intelligence is required to prevent the sentence splitter from

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2009	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Combining Different NLP Methods for HUMINT Report Analysis	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) Neuenahrer Str. 20 53343 Wachtberg-Werthhoven Germany**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADB381582. RTO-MP-IST-087 Information Management - Exploitation (Gestion et exploitation des informations). Proceedings of RTO Information Systems Technology Panel (IST) Symposium held in Stockholm, Sweden on 19-20 October 2009., The original document contains color images.

14. ABSTRACT
In this paper we present a combined approach to the automatic (pre-) analysis of intelligence reports. The combination encompasses information extraction (IE) and information enrichment by means of ontologies. The combined approach proves to yield superior results compared to standalone IE. For our work we mainly use open standards and open source software. For the purpose of IE, for instance, we use the GATE system, whereas our ontology work is based on the W3C OWL standard and the Protégé ontology editor.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**10**	

suspecting the end of a sentence after every period. Without it, a sentence would never make it past a "Mr." or "Dr." or any other abbreviation of that kind.

4. The part-of-speech-tagger that comes shipped with GATE is a rule-based tagger with a lexicon under the hood. The tagger determines the part-of-speech of the word tokens according to the categories of the Penn-Treebank tag set [16].

5. A named-entities transducer combines elements annotated by the gazetteers in step 2 above. For example, for the sequence "Dr. Mohammed el-Baradei", the gazetteer will provide the annotations *title* for "Dr.", *male forename* for "Mohammed" and *surname* for "el-Baradei" whereas a named-entity transducer uses these annotations to calculate the annotation *person* for the whole sequence.

It is, of course, essential to adapt the processing resources to the task at hand. Thus the gazetteer lists need to contain the names of towns and villages, rivers, institutions, organizations, etc. and common personal names that are relevant for the situational context.

After the aforementioned standard steps in the data processing pipeline have been completed, we next need to determine the actions, events and situations reported in the text and assign semantic roles to their participants. The expression of actions, events and situations is the domain of the verbal vocabulary, i.e. of verbs and to some degree also of deverbal nouns (nouns derived from verbs). To determine the verb and the other constituents in a sentence, we use a shallow parsing approach. Up to this point, we essentially follow the lead of Hecking and his developments for report analysis in the field of IE as implemented in the ZENON system [10].

The alternative to shallow parsing is a deep syntactic analysis. We will give a short overview over the pros and cons of deep and shallow parsing to justify our decision. Shallow parsing means that the top level constituents of a sentence are determined by means of certain (statistical or rule-based) heuristics directly from the word sequence. It indicates verb compounds, noun phrases and prepositional phrases but it yields only limited information on the internal structure of these constituents. The major advantages of shallow parsing are its robustness to unseen words and possible ambiguities, its ability to provide at least partial results even if a full analysis is not feasible, and its speed.

Deep parsing requires that for each sentence the entire syntactic structure has to be calculated. On the basis of these structures, the constituents of the sentences can then be determined. Deep parsing produces much more information than shallow parsing (in fact, more than we need for our purpose). But there are two main problems with deep syntactic analysis: unknown words and ambiguity. Additionally, deep parsing is computationally very resource-intensive. Nevertheless, within the context of the work on Hecking's ZENON system, an approach is being developed and implemented that uses a deep parser to calculate syntactic structures of report sentences and use these structures to assign semantic annotations (cf. [17] for details on this approach).

In order to illustrate the two approaches, let us take a look at an example sentence from [6] "*The wealthy widow drove an old Mercedes to the church*". Under the deep approach, the sentence structure shown in Figure 1 is calculated. It has the constituents "*the wealthy widow*", "*an old Mercedes*", and "*to the church*", and the verb "*drove*". These constituents get the roles "agent", "theme", and "destination", respectively. In the shallow approach, the same constituents ideally are determined and get assigned the same roles without calculating the sentence structure.

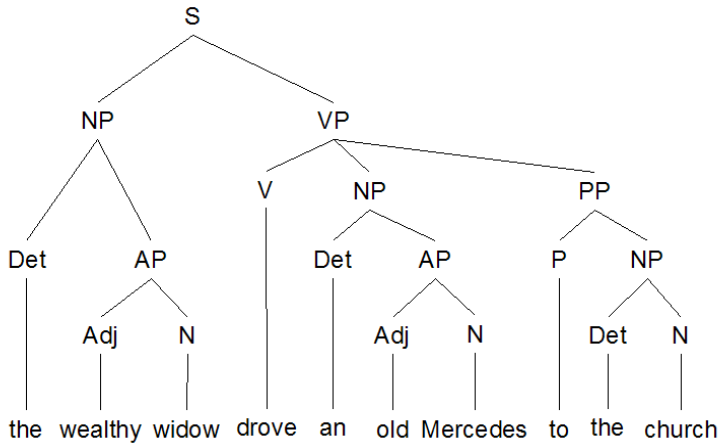

Figure 1: The complete parse tree of "The wealthy widow drove an old Mercedes to the church."

The differences between deep and shallow parsing become clearer if one examines what can go wrong with them. As mentioned above, there are two major problems for the deep approach. First, there might be a word in a sentence that has neither an entry in the lexicon nor in the named entity lists. "Mercedes" might be such a word. A word that is not recognized does not receive a category annotation. In this case, deep parsing fails completely. Second, deep parsing might come up with not only one but multiple valid sentence structures. In this case, one of them has to be chosen before constituents can be determined and thematic roles can be assigned to the constituents. Shallow parsing avoids these problems at least to some degree. It only operates with those constituents that can be determined directly. If there is an unknown word, some parts of the sentence in question will not be analyzed at all, but the rest will still be treated. Thus, in the case of unknown words, shallow parsing fails only partially. In the case of multiple structures, shallow parsing might result in multiple thematic role assignments for some of the constituents but other constituents may remain unaffected.

However, when deep parsing succeeds in calculating the sentence structure, this structure not only determines the constituents but also helps to assign the correct role. Shallow parsing has to rely more on local hints for constituent determination and role assignment. This becomes clear if we take a look at the example's twin sentence "*The wealthy widow gave an old Mercedes to the church*". Under shallow parsing, the role "destination" will be assigned to the constituent "*to the church*" due to a specific recognizer (transducer) that recognizes the preposition "*to*" and the facility "*church*". There is nothing but the verb in the second sentence that indicates that "church" in that sentence does not denote a facility but rather an organization, with the result that not "destination" but rather "recipient" is the appropriate role for "*to the church*" in this case.

3. METHOD

In the following, we present our approach, which combines shallow parsing techniques with a specific ontology. Our ontology bears characteristics of a lexical resource with a focus on the verbal lexicon. It provides information on verbs and their semantic frames [8] that enables us to enrich the results of the shallow parsing such that we can assign proper semantic roles to the verbal complements [19].

Other approaches try to exploit lexical resources more directly. Palmer and her colleagues use VerbNet (cf. [4], based on [14]) to build up their "Proposition Bank" [18, 3], whereas Lönneker-Rodman and Baker have developed a machine learning system based on FrameNet [1] for the task of Automatic Semantic Role Labelling (ASRL) [15]. FrameNet has influenced the construction of our ontology as have the works of Helbig [11] and Sowa [19].

Lönneker-Rodman and Baker in [15], subsection 5.1.2, provide a comparison between their work and the Proposition Bank which shows clear differences between these two lexical resources concerning not only their coverage but also their characteristics. These differences are explained by the fact that the Proposition Bank operates on text from the Wall Street Journal whereas the FrameNet model operates on prose texts, such as, Arthur Conan Doyle's "The Hound of the Baskervilles". This illustrates the importance of the domain for which lexical resources are being developed. We consider this a validation of our choice to develop a specific lexical resource tailored to our needs.

Our lexical resource is an OWL-based ontology [12] with a focus on verbs and their complements. It provides the kind of knowledge we need in order to assign correct semantic roles to the constituents of a sentence. Most other ontologies concentrate on the objects in the domain of interest (cf. [20]), whereas for us, the focus is on actions and these are represented as verbs in natural language. Our ontology structures the verbal vocabulary into classes with common semantic features.

The actions are divided into classes, which in turn are defined by semantic frames. This means more or less that actions belonging to the same class share the thematic roles they demand and allow. It must be mentioned, however, that a strict inheritance hierarchy with respect to the verbs' semantic features is not the goal of this ontology. But practice has taught us that verbs expressing a similar semantic concept do share a large portion of their semantic features.

The top level classes of the branch composed of verbs of our ontology form a hierarchy similar to the one proposed in [11]. The topmost class is *Situation*; all situations have in common that they can be located in time and space. As a result, the properties *when* and *where* are already defined at this level and propagate down to each individual verb. Situations are divided into *Dynamic Situations* and *Static Situations*. Dynamic situations comprise verbs expressing that something is going on, while static situations are verbs expressing states in a wide sense. Dynamic situations are further subdivided into *Actions* and *Events*. Actions are characterized by the fact that they are performed by an *agent*, whereas events do not feature such an agent. The verbs *happen* or *occur* are typical representatives of events. The vast majority of verbs from our corpus belong to the *Action* class.

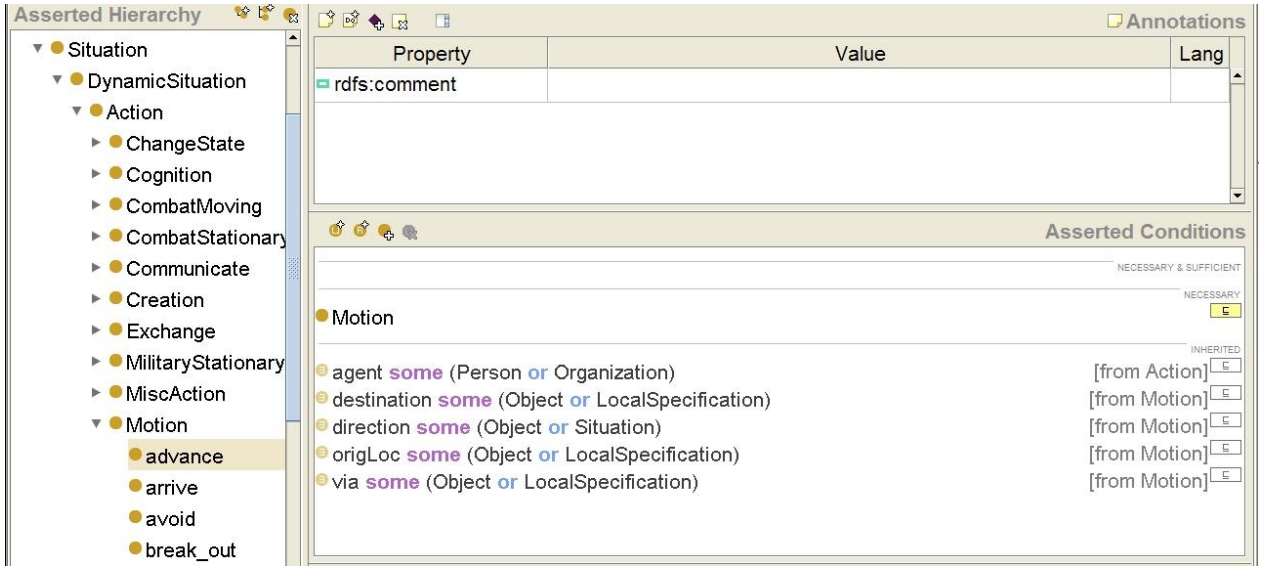

Figure 2: This snippet from a Protégé screen shows the semantic properties of the verb *advance*.

Consider, for example, the verb *advance* for which the ontological entry is shown in Figure 2. An *advance* action demands an *agent* as well as a *direction* or a *destination*. Direction and destination are spatial roles

that correspond to spatial constituents as annotated by the information extraction step. So, ontological information about semantic frames of actions – together with other more standard constraints represented in the ontology – enables us to map constituents to roles. Spatial constituents, of course, are mapped to spatial roles while the prepositions that start the spatial constituents indicate the correct role. For example, "towards" indicates a direction and "to" a destination, whereas "from" indicates a spatial origin (*origLoc*), a thematic role optional for an advance action. Similarly, temporal constituents can be mapped to temporal roles (start, duration, completion, point in time; (cf. [19], p. 508). With respect to our example of the "wealthy widow" twin sentences, the ontology provides entries for the verbs "drive" and "give". "Drive" is a verb from the *Motion*-class and thus in most respects similar to "advance". Therefore, a prepositional phrase starting with the preposition "to" matches the requirements of the *destination* slot of "drive" such that the phrase "to the church" receives *destination* as semantic annotation in the "drive"-sentence. In contrast, "give" is a verb of the *Exchange*-class and has *agent*, *recipient*, and *affected* as its associated thematic roles. Therefore in this case, the prepositional phrase "to the church" matches *recipient*, which thus becomes the semantic annotation of the phrase.

4. EXAMPLES

The method for text analysis as described above is used as a component in a system for automatic threat recognition which is under development. The development of this system is led by the German company IABG. Below, we will describe how the text analysis component is integrated into the threat recognition system and we will discuss examples to better illustrate how it works.

In general, the threat recognition system first stores incoming HUMINT reports in English or German in its report database [13]. A database entry – a report in the database – consists of four parts. The first part contains "header" information, such as sender and recipient, date/timestamp, security estimation, and how the sender judges the report's (and its source's) reliability and credibility. The second part stores topic information, which may be added to the report later by, for example, the sender or the recipient. The third part is the report content itself . "Content" here means the content as contained in the original (i.e., unexpanded) report. This content will not change during the processing and thus serves as a reference whenever a user of the system wants to check the results of the analysis against the original reports. The fourth part is the formal representation of the report's content. The formal representation is a result of analysis and can be changed and modified interactively. In order to produce the first instance of that formal representation, the method discussed in section 3 is applied.

As soon as there is any (partial) result in the formal representation slot, the report can be used for threat recognition analysis. In order to run the threat recognition analysis process, the user activates a section of the system's threat model which has been developed by IABG using knowledge collected by IABG staff during a six month stay at the Bundeswehr camp in Kundus, Afghanistan. The user activates that part of the threat model which she is interested in. This part then activates its corresponding indicators. Indicators [12] are entities which have been previously specified by experts. An indicator has the same structure as the data produced during the analysis process. The main difference between these two types of data is that indicators are underspecified with respect to certain features. An indicator then matches those pieces of data form the analysis that can be unified with the indicator.

In order to illustrate the effect that our text analysis method has on the whole process, we present some simple examples in the following. Real examples are much more complex since more interdependencies are involved. However, since this presentation concentrates on text analysis, we consider simple examples more illustrative. Let us assume that the user wants to check whether the data indicates a threat against the camp of her unit. She then would activate the respective section of the threat model. This activates the corresponding indicators. One such indicator, for example, is triggered by *sensors found inside the camp*. The logic behind this indicator is that an unknown sensor inside the camp indicates that someone is spying

on the camp, i.e., the camp is in danger. Say the camp is called “Camp Pilsen”. Then the indicator would have the verb “find”, and the semantic roles “theme” and “location”. “Theme” is filled by “object of type sensor” and “location” is filled by “Camp Pilsen”. The other semantic role slots that are, according to the ontology, part of the frame of “find”, namely “agent” and “Point in Time”, do not have a filler in that indicator. The indicator will become active whenever a sensor is found at the camp, regardless of who finds it or when it is found (in other words, this latter information in insignificant for this indicator). This is what is meant by our earlier statement that the indicator is an *underspecified* data structure. If there is a report in the database that has an entry saying that “Corporal Zirndorf found an A-sensor inside Camp Pilsen an hour ago” the indicator would match because the report’s fillers of all roles match the fillers as given by the indicator.

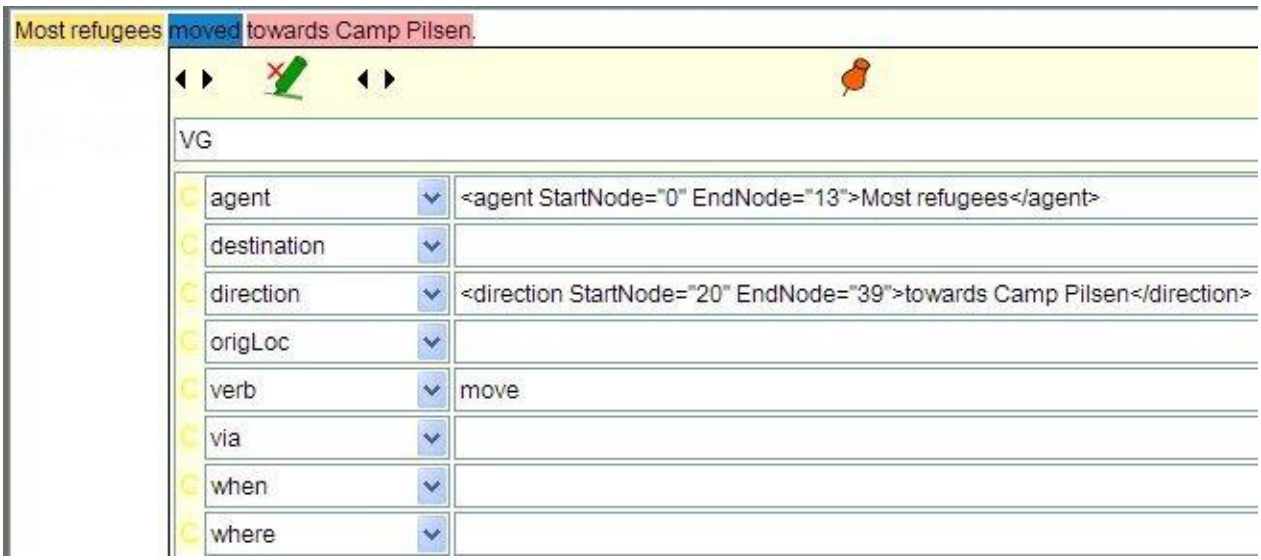

Figure 3: Snapshot showing the analysis results for “Most refugees moved towards Camp Pilsen”

The value of the system depends on all of its subsystems. Of course, the threat model has to be precise enough to be of use as do its indicators. Report analysis also is crucial. Insufficient analysis will result in underspecified data structures. Combined with underspecified indicators, too many matches may occur. Let us assume, for example, that we have an indicator saying the camp is endangered if hostile persons or forces move towards the camp. Thus, the indicator has the verb “move” which is a verb of the *Motion*-class. The roles that are filled are *direction* (filled by “Camp Pilsen”) and *agent*. The indicator does not make any further assumptions about the agent except that it has to be an entity classified as *hostile*. Now let us take a report saying “Most refugees moved towards Camp Pilsen”. Here, “Most refugees” fills the agent slot and “towards Camp Pilsen” fills the destination slot. Figure 3 shows the respective snapshot of the text analysis component’s display. Obviously, the verb of the report sentence is identical to the indicator verb. The same holds for the fillers of “direction” in the report sentence and in the indicator. However, the refugees might be classified as “neutral” and not as “hostile” by ontological means. Thus, no match occurs.

Figure 4: Snapshot for "Most refugees moved from Friedland towards Camp Pilsen"

If we have a more elaborated report "Most refugees moved from Friedland towards Camp Pilsen", the "origLoc" slot of the report is also filled as can be seen in figure 4. However, since the indicator is underspecified with respect to this slot, this does not affect the indicator match. Again no match occurs.

Figure 5: Snapshot for "Most refugees from Friedland moved towards Camp Pilsen"

Now, we try the report "Most refugees from Friedland moved to Camp Pilsen". The prepositional phrase "from Friedland" in this sentence is attached to "most refugees" which means that the refugees originate from Friedland. Formally, thus, the report does no longer say that the movement of the refugees originated in Friedland. However, this is not the problem. The problem is that under our shallow parse, "most refugees" is still classified as noun phrase and "from Friedland" still is classified as prepositional phrase. But the two phrases are not combined into one noun phrase. As a result, there is no noun phrase anymore directly in front of the verb. It turns out this is one of the ways in which the current version of our text analysis component determines an "agent". As a result, the agent slot is not filled for the formal representation of the report, cf. figure 5, and thus the report representation is underspecified with respect to the agent role.

At this point, a design decision had to be made. As the agent slot was not filled, we might have decided that the report did not provide enough content (in the form of filled slots) to be considered for matching. In this case, there would have been no match and thus no alarm. The other alternative would have been to allow the match. Since the agent slot was not filled, there was a possibility that the (unknown) agent of the moving action towards the camp was hostile. In this case the match would have been successful and an alarm would have occurred. The design decision depends on the use of the threat recognition system. In one case, we receive fewer alarms but all the alarms which occur are based on reports which had been analysed to a sufficient degree to match all restrictions of the respective indicators. In the other case, we receive more alarms but most of them are based on matches between indicators and underspecified report representations.

5. CONCLUSIONS

In this paper, we have presented a method to analyse HUMINT reports written in natural language. The method uses shallow information extraction techniques based on GATE. We alleviate the disadvantages of the shallow approach by using ontological knowledge about verbs and their semantic frames. The verbs and frames under consideration are taken from the HUMINT domain. The frame information attached to a verb constrains the semantic roles that can be assigned to the sentence's constituents.

The method presented for report analysis can be a component of larger systems, e.g. machine translation systems that translate reports into all languages being used in a complex combined operation, or systems for analyzing large numbers of reports under specific questions. In this paper, we have sketched how the report analysis component operates in a system for automatic threat recognition.

6. REFERENCES

[1] FrameNet. http://framenet.icsi.berkeley.edu/.

[2] GATE: A General Architecture for Text Engineering. http://gate.ac.uk/.

[3] PropBank: The Proposition Bank. http://verbs.colorado.edu/ mpalmer/projects/ace.html.

[4] VerbNet: A Class-Based Verb Lexicon. http://verbs.colorado.edu/ mpalmer/projects/verbnet.html.

[5] Douglas E. Appelt and David J. Israel. Introduction to Information Extraction Technology. A tutorial prepared for IJCAI-99, Stockholm, Sweden, 1999.

[6] J. Kathryn Bock and Helga Loebell. Framing Sentences. *Cognition*, 35(1): 1–39, 1990.

[7] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[8] Charles J. Fillmore. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart And Winston, New York, 1968.

[9] Matthias Hecking. Information Extraction from Battlefield Reports. In *Proceedings of the 8th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington, DC, U.S.A., 2003.

[10] Matthias Hecking. System ZENON. Semantic Analysis of Intelligence Reports. In *Proceedings of the LangTech 2008*, Rome, Italy, February 28 – 29, 2008.

[11] Hermann Helbig. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, 2006.

[12] Constantin Jenge and Miloslaw Frey. Ontologies in Automated Threat Recognition. In *Proceedings of the Military Communications and Information Systems Conference (MCC) 2008*, Cracov, Poland, September 23 – 24, 2008.

[13] Jürgen Kaster, Wolf-Dieter Huland, and Sascha Huy. Dokumentenverwaltung in der G2-Datenbank Einsatz (v2.0). Technical Report, Forschungsgesellschaft für Angewandte Naturwissenschaften e.V., Wachtberg, Germany, 2003.

[14] Beth Levin. *English Verb Classes And Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL, 1993.

[15] Birte Lönneker-Rodman and Collin F. Baker. The FrameNet Model and its Applications. *Natural Language Engineering*, 15(3):415 – 453, 2009.

[16] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313 – 330, 1993.

[17] Sandra Noubours. Annotation semantischer Rollen in HUMINT-Meldungen basierend auf dem statistischen Stanford-Parser und der lexikalischen Ressource VerbNet. Master's thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2009.

[18] Martha Palmer, Dan Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):75 – 105, 2005.

[19] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.

[20] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.