Multilingual Content Extraction
Extended with Background Knowledge for Military Intelligence

Dr. Matthias Hecking
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Neuenahrer Straße 20, 53343 Wachtberg, Germany
matthias.hecking@fkie.fraunhofer.de

Dr. Andreas Wotzlaw
University of Cologne, Department of Computer Science
Pohligstrasse 1, 50969 Köln, Germany
wotzlaw@informatik.uni-koeln.de

Ravi Coote
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Neuenahrer Straße 20, 53343 Wachtberg, Germany
ravi.coote@fkie.fraunhofer.de
Multilingual Content Extraction Extended with Background Knowledge for Military Intelligence

Matthias Hecking¹, Andreas Wotzlaw², and Ravi Coote³

¹ Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Neuenahrer Str. 20, 53343 Wachtberg, Germany
{matthias.hecking, ravi.coote}@fkie.fraunhofer.de

² Universität zu Köln, Institut für Informatik
Pohligstrasse 1, 50969 Köln, Germany
wotzlaw@informatik.uni-koeln.de

Abstract. Written information for military purposes is available in abundance. Documents are written in many languages. The question is how we can automate the content extraction of these documents. One possible approach is based on shallow parsing (information extraction) with application-specific combination of analysis results. The ZENON research system is an example, it does a partial content analysis of some English, Dari and Tajik texts. Another principal approach for content extraction is based on a combination of deep and shallow parsing with logical inferences on the analysis results. In the project ”Multilingual content analysis with semantic inference on military relevant texts” (mIE) we followed the second approach. In this paper we present the results of the mIE project. First, we briefly contrast the ZENON project to the mIE project. In the main part of the paper, the mIE project is presented. After explaining the combined deep and shallow parsing approach with Head-driven Phrase Structured Grammars, the inference process is introduced. Then, we show how background knowledge is integrated into the logical inferences to increase the extent, quality and accuracy of the content extraction. The prototype is also presented.

1 Introduction

The new deployments of the German Federal Armed Forces (Bundeswehr) cause the necessity to analyze large quantities of intelligence reports and other documents written in different languages. Especially the content analysis of free-form texts is important for any information operation. During the content analysis the actions described and entities involved are extracted from the texts, combined (fused), enhanced with background knowledge and stored for further processing. A partial content analysis can be created through information extraction (IE) which is a natural language processing technique (see [AI99], [Hec03b], [Hec04b]). In our ZENON project (see [HS08], [Hec09], [HB10]) we use this shallow parsing approach to realize the partial content analysis.

Multilingual information extraction is a current research topic (see [PS07]). The main idea of multilingual information extraction is the extraction of information about a specific entity and/or action from documents written in different languages. If information written in different languages can be (partially) extracted and fused automatically - without the use of a human translator - this would speed up the information gathering and combining process. This would also be the case if the performance of the information extraction for the different languages is developed differently.

In the project ”Multilingual content analysis with semantic inference on military relevant texts” (mIE), we extended the basic ideas of the ZENON project in two ways. First, the shallow parsing approach is extended to a combined deep and shallow parsing approach. The extracted meaning of each sentence is formalized in formal logic. Simple English and Arabic texts can be processed. Second, the formalized content is extended with background knowledge (WordNet [Fel98], YAGO [SKW08]) so that new conclusions (logical inferences) can be drawn. For this purpose theorem provers and model builders are used.

The overall objective of the mIE project is to demonstrate that it is possible to use state-of-the-art natural language processing techniques to extract and combine military relevant knowledge from free-form texts even for rare languages. An expected advantage of systems like mIE is the increased productivity of the intelligence analyst. He might analyze and combine information from more intelligence reports and
The approaches for content extraction can be classified coarse-grained according to two dimensions. The first dimension characterizes how deeply the syntactic/semantic analysis is performed. Possible types along this dimension could be: shallow parsing, combined shallow and deep parsing and deep parsing. The second dimension characterizes how the results of the analysis are used further. Possibilities are here: application-specific combination of the analysis results, general combination of the analysis results through logical inferences or use of formal represented background knowledge.

The first approach for content extraction which is used in most of the current content extraction projects can be characterized by: shallow parsing with application specific combination of analysis results. The used parsing technique is based on IE (see [AI99]). Our ZENON system is an example for this approach.
The second approach for content extraction can be characterized by: combined deep and shallow parsing with logical inferences on the analysis results and on background knowledge. Our mIE project is an example for this approach.

**Project ZENON.** To understand the differences of the two approaches more clearly a short overview of the ZENON research project is given. In ZENON (see [Hec03a], [Hec03b], [He04a], [He04b], [He06a], [He06b], [He06c], [He07], [Sch07], [He09], [SB10], [HB10], [Nou10]) a multilingual IE approach is used for the (partial) content analysis from texts written in different languages. The ZENON system uses a shallow syntactic approach based on chunk-parsing and transducer. The approach is called ‘shallow’ because only those parts of a sentence are analyzed which are of interest for the application, e.g., if only informations about persons are of interest, then only person names, addresses, etc. are identified in the texts and processed. The main advantage of this approach is its robustness when confronted with ungrammatical sentences. The disadvantage is that relevant information may possibly be missed. The transducers are handcrafted grammars processed as finite automata.

At the moment, the ZENON system (see Fig. 1) is able to process English documents (similar in structure and vocabulary to HUMINT reports from the KFOR deployment of the Bundeswehr) and documents written in Dari. The Tajik module is not yet integrated into the prototype. The knowledge about the actions and named entities is identified from each sentence, and the content of the sentences are represented formally as typed feature structures. These formal representations can be combined and presented in a graphically navigatable Entity-Action-Network.

In the current version of the ZENON system the information extraction results from two different languages (English and Dari) are combined. Beside the information extraction, the system gives a simple word-to-word-translation for Dari (to German) to further support the analyst. This allows the analyst to access information from Dari texts without knowing these languages. The automatic processing of the texts also extends the volume of these texts the analyst can handle. In view of the limited capabilities of the available natural language processing techniques, the ZENON system is only an assistance of the analyst.

In the rest of this paper the mIE project is presented as an example of the second approach. A complete description of the ideas, concepts and the implemented prototype can be found in [HWC11], [CW10], [WC10] and [Wot10].

### 3 Combined Deep and Shallow Parsing with Logical Inferences

#### 3.1 Basic Idea of the mIE Project

In the project "Multilingual content analysis with semantic inference on military relevant texts" (mIE) information from simple documents written in different languages can be combined. A combined deep and shallow (syntax and semantic) parsing technique is used to increase the quality and accuracy of the parsing results. The meaning of each sentence is formalized in formal logic and such formalized content is extended with background knowledge (WordNet, YAGO) so that new conclusions (logical inferences) can be drawn.

Our aim is to provide a robust, modular, and highly adaptable environment for a linguistically motivated large-scale semantic text analysis.

The problem of drawing conclusions on texts and background knowledge is formalized as a pair of a text and a hypothesis. The following is a typical example:

Text $T$:

German soldiers were involved in a battle near Kundus. Two of them were badly injured. They were brought with a military airplane to Germany.

Hypothesis $H$:

Some hurt soldiers were transported to Germany.
For the automatic answer whether the hypothesis follows or not various problems have to be solved. For example, the sentences must be processed linguistically or background knowledge is necessary for the inference steps "from injure infer to hurt" and "from transport infer to bring".

Drawing inferences on military relevant texts can be formulated as a problem of recognizing textual entailment (RTE, see [DDMR09,BDD+09]). In RTE we want to identify automatically the type of a logical relation between two input texts \((T \text{ and } H)\). In particular, we are interested in proving the existence of an entailment between them. The concept of textual entailment indicates the state in which the semantics of a natural language written text can be inferred from the semantics of another one. RTE requires a processing at the lexical, as well as at the semantic and discourse level with an access to vast amounts of problem-relevant background knowledge [Bos05].

RTE is without doubt one of the ultimate challenges for any NLP system. As a generic problem, it has many useful applications in NLP [GMDD07]. Interestingly, many application settings like, e.g., information retrieval, paraphrase acquisition, question answering, or machine translation can fully or partly be modeled as RTE [BDD+09]. Entailment problems between natural language texts have been studied extensively in the last few years, either as independent applications or as a part of more complex systems (e.g., RTE Challenges [BDD+09]).

In our setting, we try to recognize the type of the logical relation between two input texts, i.e., between the text \(T\) (usually several sentences) and the hypothesis \(H\) (one short sentence). More formally, given a pair \(\{T, H\}\), our system can be used to find answers to the following, mutually exclusive conjectures with respect to background knowledge relevant both for \(T\) and \(H\) [BB05]:

1. \(T\) entails \(H\),
2. \(T \land H\) is inconsistent, i.e., \(T \land H\) contains some contradiction, or
3. \(H\) is informative with respect to \(T\), i.e., \(T\) does not entail \(H\) and \(T \land H\) is consistent.

We aim to solve a given RTE problem by applying a model-theoretic approach where a formal semantic representation of the problem, i.e., of the input texts \(T\) and \(H\), is computed. However, in contrast to automated deduction systems [Akh05] which compare the atomic propositions obtained from the text and the hypothesis in order to determine the existence of entailment, we apply logical inference of first-order. To compute adequate semantic representations for input problems, we build on a combination of deep and shallow techniques for semantic analysis. Our mIE system consists of three main modules (see Fig. 2):

1. **Syntactic and Semantic Analysis**, where the combined deep-shallow semantic analysis of the input text is performed;
2. **Logical Inference**, where the logical deduction process is implemented (it is supported by two external components with external knowledge and inference machines);
3. **Graphical User Interface**, where the analytical process is supervised and its results are presented to the user.

![Fig. 2. Main modules of the framework for semantic text analysis](image)

In order to solve a given RTE problem, the texts representing \(T\) and \(H\) go first through the syntactic processing and semantic construction where formal representations of the meaning are computed. This
task is performed by the first module of the framework (see Fig. 2). It is build on the XML-based middleware architecture Heart of Gold [Sch07a] centered around the English Resource HPSG Grammar (ERG, see [Fli00]). It allows for a flexible integration of shallow and deep linguistics-based and semantics-oriented NLP components like, e.g., the statistical part-of-speech tagger TnT [Bra00], the named entity recognizer SProUT [DKP+04], or the deep HPSG parser PET [Cal00]. See Section 3.2 for more details.

The main problem with approaches processing text in a shallow fashion is that they can be tricked easily, e.g., by negation, or systematically replacing quantifiers. Also an analysis solely relying on some deep approach may be jeopardized by a lack of fault tolerance or robustness when trying to formalize some erroneous text (e.g., with grammatical or orthographical errors) or a shorthand note. The main advantage when integrating deep and shallow NLP components is increased robustness of deep parsing by exploiting information for words that are not contained in the deep lexicon [Sch07a]. The type of unknown words can then be guessed, e.g., by usage of statistical models.

The semantic representation language used for the results of the deep-shallow analysis is a first-order fragment of Minimal Recursion Semantics (MRS, see [CFPS05]). However, for their further usage in the logical inference, the MRS expressions are translated into another, semantic equivalent representation of First-Order Logic with Equality (FOLE) [BB05]. This logical form with a well-defined model-theoretic semantics was successfully applied for RTE in [CCB07].

As already mentioned, an adequate representation of a natural language semantics requires access to vast amounts of common sense and domain-specific world knowledge. RTE systems need problem-relevant background knowledge to support their proofs (see [Bos05] and [BM06]). The logical inference in our system (performed in the second module) is supported by external background knowledge integrated automatically and only as needed into the input problem in form of additional first-order axioms. In contrast to already existing applications (see, e.g., [CCB07],[BDD+09]), our system enables flexible integration of background knowledge from more than one external source (see Section 3.4 for details).

The ideas of the mIE project were realized in a research prototype. In Fig. 3 the GUI (Graphical User Interface) with an example of $T$ and $H$ is shown. We have also build a simple HPSG Arabic grammar, so our system is able to process simple Arabic sentences, too. In Fig. 4 $T$ consists of sentences in Arabic and $H$ in English.
3.2 Deep-shallow Semantic Text Analysis

After entering the system via the user interface the texts go first through the syntactic processing and semantic construction of the first system module. To this end, they are analyzed by the components of the XML-based middleware architecture Heart of Gold (see Fig. 5). It allows for a flexible integration of shallow and deep linguistics-based and semantics-oriented NLP components, and thus constitutes a sufficiently complex research instrument for experimenting with novel processing strategies. Here, we use its slightly modified standard configuration for English centered around the English Resource HPSG Grammar (ERG, see [Fli00]). The shallow processing is performed through statistical or simple rule-based, typically finite-state methods, with sufficient precision and recall. The particular tasks are realized as follows: the tokenization task with the Java tool JTok, the part-of-speech tagging with the statistical tagger TnT [Bra00] trained for English on the Penn Treebank [MMS93], and the named entity recognition with SProUT [DKP+04]. The latter one, by combining finite state and typed feature structure technology, plays an important role for the deep-shallow integration, i.e., it prepares the generic named entity lexical entries for the deep HPSG parser PET [Cal00]. This makes sharing of linguistic knowledge among deep and shallow grammars natural and easy. PET is a highly efficient runtime parser for unification-based grammars and constitutes the core of the rule-based, fine-grained deep analysis. The integration of NLP components is done either by means of an XSLT-based transformation, or with the help of Robust Minimal Recursion Semantics (RMRS, see [Cop03]) when a given NLP component supports it natively.

Minimal Recursion Semantics (MRS). MRS is the formal description of the meaning of sentences. In this formalism scope underspecification is used. It is a well-known technique in computational semantics of natural language [Bum07]. MRS is a description language over formulas of FOL languages with generalized quantifiers. For instance, the sentence “Every wizard acts in a circus” illustrates the well-known problem of scopal ambiguity. Is it one and the same circus in which every wizard acts or are there possibly several different circuses in which the wizards act? Thus, the sentence has two scopal readings which are represented by FOL formulas. MRS allow multiple formulas, which differ only in their scopal configuration to be expressed with exactly one single compact formula.

Robust Minimal Recursion Semantics (RMRS). RMRS is a generalization of MRS. It can not only be underspecified for scope as MRS, but also partially specified, e.g., when some parts of the text cannot be resolved by a given NLP component. Furthermore, in RMRS due to possible lack of morphological analysis, predicates are allowed to lack for their arguments. Hence, it can be used as a semantic representation formalism of shallow NLP components. HOG supports integration of shallow NLP components by using RMRS as an exchange format.
Furthermore, RMRS is a common semantic formalism for HPSG grammars within the context of the *LinGO Grammar Matrix* [BFO02]. Besides ERG, which we use for English, there are also grammars for other languages like, e.g., the Japanese HPSG grammar *JaCY* [SB02], the *Korean Resource Grammar* [JBJ05], the *Spanish Resource Grammar* (SRG, see [Mar02]), or the proprietary German HPSG grammar [CZ09]. Since all of those grammars can be used to generate semantic representations in form of RMRS, a replacement of ERG with another grammar in our system can be considered and thus a high degree of multilinguality is achievable.

The combined results of the deep-shallow analysis in RMRS form are transformed into MRS and resolved with UTool 3.1 [KT05]. UTool enumerates all text readings (resolving RMRS) and this enumeration is passed on to the logical inference.

Texts written in two different languages (English, Arabic) are analyzed. In Fig. 6 the result of the deep analysis of an Arabic sentence is shown. In Fig. 7 an example of a semantic representation as MRS is given.

### 3.3 Logical Inferences on Text Content

The results of the semantic analysis in form of MRS are sent to the module for logical inference (see Fig. 8), where they are translated into another, semantic equivalent representation of *First-Order Logic with Equality* (FOLE). This logical form with a well-defined model-theoretic semantics was already applied for RTE (see [BB05],[CCB07]).

An adequate representation of natural language semantics requires an access to a vast amount of common sense and domain-specific knowledge. As already clearly indicated in [BM05], RTE systems need problem-relevant background knowledge to support their proofs. Unfortunately, the existing applications today use typically only one source of background knowledge, e.g., WordNet or Wikipedia. They could
Fig. 6. Deep arabic analysis

Fig. 7. Semantic representation as MRS
Fig. 8. Logical inference with external inference machines and background knowledge

boost their performance if a huge ontology with knowledge from several sources would be available. Such knowledge base would have to be of high quality and accuracy comparable with that of an encyclopedia. It should include not only ontological concepts and lexical hierarchies like those of WordNet, but also a great number of named entities (here also referred to as individuals) like, e.g., people, geographical locations, organizations, events, etc. Also other semantic relations between them, e.g., who-was-born-when, which-language-is-spoken-in, etc. should be comprised (factual knowledge). Here, we mean by ontology any set of facts and/or axioms comprising potentially both individuals (e.g., Berlin) and concepts (e.g., city).

To this end, the module for logical inference supports integration of external knowledge sources and by using them it extends automatically the locally stored FOLE formulas with problem-relevant knowledge in form of background knowledge axioms (see Sect. 3.4).
We integrated two huge sources of external knowledge. WordNet 3.0 [Fel98] is used as a lexical database for synonymy, hyperonymy (e.g., 'location' is a hypernym of 'city'), and hyponymy (e.g., 'city' is a hyponym of 'location') relations (approx. 2.6 million entries). It helps the logical inference process to detect entailments between lexical units from the text and the hypothesis. It serves also as a database for individuals but a very small one if compared to the second source. For efficiency purposes, it was integrated directly into the module. Conceptually, the hyperonymy/hyponymy relation in WordNet spans a directedacyclic graph (DAG) with the root node entity[see [Fel98], [SKW08]]. This means that there are nodes representing various concepts or individuals in the WordNet graph that are direct hyponyms of more than one concept. For that reason, the knowledge axioms which are generated later from the WordNet graph may induce inconsistencies between the input problem formulas and the extracted knowledge. This can be very harmful for the subsequent logical inference process. In Sect. 3.4 we discuss this problem in more detail and present several strategies that can deal with this restriction.

YAGO [SKW08], the second source we use, is a large and arbitrarily extensible ontology with high precision and quality (approx. 22 million facts and relations). Its core was assembled automatically from the category system and the infoboxes of Wikipedia, and combined with taxonomic relations from WordNet. Similar to WordNet, the concepts and individuals hierarchy of YAGO spans a DAG. Thus, we must proceed carefully when integrating data from that source into the RTE problem, too (see Sect. 3.4). To access YAGO, we use a dedicated query processor with its own query language, similar to that of [SKW08]. The query processor first normalizes the shorthand notation of the query, and after translating it into SQL, sends it to the MySQL-Server. The query results are first preprocessed by the query processor, so that only those concepts are sent back for integration which are consistent with WordNet concept hierarchy.

After the computation of relevant background knowledge and its integration into the input, the resulting extended RTE problem is solved by the inference process (see Fig. 8). To check which logical relation for the extended RTE problem holds (whether the logical relation is an entailment, a contradiction, or informative), we use external automated reasoning tools like finite model builders (e.g., Mace4 [McC03]) and theorem provers (e.g., Prover9 [McC09]). While theorem provers are designed to prove that a formula is valid (i.e., the formula is true in any model), they are generally not good at deciding that a formula is not valid. Model builders are designed to show that a formula is true in at least one model. The experiments with different inference machines show that solely relying on theorem proving is in most cases insufficient due to low recall. Indeed, our inference process incorporates model building as a central part of the inference process. Similar to [CCB07], we exploit the complementarity of model builders and theorem provers by applying them in parallel to the input RTE problem in order to tackle with its undecidability more efficiently. More specifically, the theorem prover attempts to prove the input whereas the model builder simultaneously tries to find a model for the negation of the input.

In Fig. 9, an example of a FOLE formula produced from MRS is shown. Fig. 10 presents a result of the inference process. In this case the hypothesis $H$ is entailed from the text $T$. 
3.4 Background Knowledge

In the following we describe our two-phase integration procedure which we apply for the integration of ontological knowledge from two sources, WordNet and YAGO, into the logical inference process of RTE. In particular, we show how we can combine problem-relevant individuals and concepts from YAGO with those from WordNet so that the consistency of background knowledge axioms is preserved whereas the original logical properties of the input RTE problem do not change. Since the input problem itself may be consistent and our goal is to prove it, the knowledge we integrate into it must not make it inconsistent.

To make our presentation as comprehensible as possible, we apply our procedure to a small RTE problem which we augment with relevant background knowledge axioms in the course of this section. More specifically, we want to prove that the text \( T \):

\[
\text{Leibniz was a famous German philosopher and mathematician born in Leipzig. Thomas reads his philosophical works while waiting for a train at the station of Bautzen.}
\]

entails the hypothesis \( H \):

\[
\text{Some works of Leibniz are read in a town.}
\]

In order to prove the entailment above, we must know, among other things, that \( Bautzen \) is a town. We assume that no information about \( Bautzen \), except that it is a named entity (i.e., an individual), were yielded by the deep-shallow semantic analysis. However, we expect that this missing information can be found in the external knowledge sources. The search for relevant background knowledge begins after the first-order representation of the problem is computed and translated into FOLE (see Fig. 8). At this stage, the RTE problem has already undergone syntactic processing, semantic construction, and anaphora resolution in our framework which together have generated a set of semantic representations of the problem in form of MRS.

The integration procedure is composed of two phases. In the first phase we search for relevant knowledge in WordNet, whereas in the second phase we look for additional knowledge in YAGO which we combine afterwards with that found in the first phase. Finally, we generate from the knowledge we have found and successfully combined background knowledge axioms and integrate them into the set of FOLE formulas representing the input RTE problem.

First Phase: Integration of WordNet. At the beginning of the phase, we list all predicates (i.e., concepts and individuals) from the input FOLE formulas. They will be used for the search in WordNet. In the implementation we consider as search predicates all nouns, verbs, and named entities, together with their sense information which is specified for each predicate by the last number in the predicate name, e.g., sense 2 in \textit{work,n,2}. In WordNet, the senses are generally ordered from most to least frequently
used, with the most common sense numbered 1. Frequency of use is determined by the number of times a sense was tagged in the various semantic concordance texts used for WordNet [Fel98]. Senses that were not semantically tagged follow the ordered senses. For our small RTE problem we can select as search predicates, e.g., work_n_2, read_v_1, or leibniz_per_1. It is important for the integration that the sense information computed during the semantic analysis matches exactly the senses used by external knowledge sources. This ensures that the semantic consistency of background knowledge is preserved across the semantic and logical analysis. However, this seems to be an extremely difficult task, which does not seem to be solved fully automatically yet by any current word sense disambiguation technique. Since in WordNet but also in ERG the senses are ordered by their frequency, we take for semantic representations generated during semantic analysis the most frequent concepts from ERG.

Having identified the search predicates, we try to find them in WordNet and, by employing both the hyperonymy/hyponymy and synonymy relations, we obtain a knowledge graph $G_W$. A small fragment of such a knowledge graph for text $T$ of our example is given in Fig. 11. In general, $G_W$ is a DAG with leaves represented by the search predicates, whereas its inner nodes and the root are concepts coming from WordNet. The directed edges in $G_W$ correspond to the hyponym relations, e.g., in Fig. 11, the named entity leipzig is a hyponym of the concept city. Note that in the opposite direction they describe the hyperonym relations, e.g., the concept city is a hyperonym of the named entity leipzig. Each synonymy relation is represented in $G_W$ by a complex node composed of synonymous concepts induced by the relation (i.e., all concepts represented by a complex node belong to the same synset in WordNet), e.g., the complex node with concepts district and territory in Fig. 11.

![Fig. 11. Fragment of knowledge graph $G_W$ after the search in WordNet](image)

Furthermore, it can be seen in Fig. 11 that the leaf representing individual leipzig has more than one direct hyperonym, i.e., there are three hyponym relations for leaf leipzig with concepts administrative_district, city, and planet. This property of graph $G_W$ may cause inconsistencies when the background knowledge axioms are later generated from it and integrated into the input FOLE formulas.

The graph $G_W$ is optimized so that only those concepts from $G_W$ appear in the new tree $T_K$, generated from $G_W$, which are directly relevant for the inference problem. Thus, all knowledge which will not add any inferential power is removed. For a complete description of the optimization process see [Wot10].

One can see in Fig. 12 that not all search predicates were recognized enough precisely during the first phase. More specifically, the named entity bautzen was not classified as a town as we would expect that. Since a suitable individual was not found in WordNet, the named entity bautzen_ne_1 was assigned
Second Phase: Integration of YAGO. In this phase we consult YAGO about search predicates that were not recognized in the first phase. We formulate for each such predicate an appropriate query and send it to the query processor. To this end, we use relation type, one of the build-in ontological relations of YAGO [SKW08]. For our small RTE problem, we ask YAGO with a query bautzen type ? of what type (or in YAGO nomenclature: of what class) the named entity bautzen is. If succeed, it returns knowledge graph $G_Y$ with WordNet concepts which classify the named entity. Fig. 14 depicts graph $G_Y$ for our example. We can see that bautzen was now classified more precisely, among other things, as a town.

In general, each graph $G_Y$ is a DAG composed of partially overlapping paths leading (with respect to the hyperonymy relation) from some root node (i.e., the most general concept in $G_Y$, e.g., node object in Fig. 14) to the leaf representing the search predicate (e.g., the complex node bautzen in Fig. 14). Observe that there is one and only one leaf node in every graph $G_Y$. Since the result of every YAGO-query is in general represented by a DAG, we cannot integrate it completely into the knowledge tree $T_K$. According to the leaf of $G_Y$ in Fig. 14, the named entity bautzen can also be classified as an asteroid or an administrative district.

In order to preserve the correctness of results, we select for the integration into tree $T_K$ only those concepts, individuals, and relations from $G_Y$ which lay on the longest path from the most general concept in $G_Y$ to one of the direct hyperonyms of the leaf, and which has the most common nodes with the knowledge tree $T_K$ from the first phase. In Fig. 14 the concepts and individuals on the gray shaded path were chosen by our heuristic for the integration into $T_K$. After the path has been selected, it is optimized and integrated into the knowledge tree $T_K$. Fig. 15 depicts the knowledge tree $T_K$ after the gray shaded path from Fig. 14 was integrated into it.

Observe finally that the integration of selected parts of graph $G_Y$ into tree $T_K$ is performed sequentially for each search predicate which was not classified in the first phase (note that each search generates its own knowledge graph $G_Y$).

Additionally to the first query to YAGO, we can also formulate a second one like bautzen isCalled ?, in which we ask what are the names of the named entity in other languages. In Fig. 14 we can see four different names for this entity. This complementary information can be combined afterwards into the FOLE formulas of the RTE problem as new predicates, e.g.,

$$... \exists x ((\text{bautzen}(x) \leftrightarrow \text{budysin}(x) \leftrightarrow \text{budissa}(x) \leftrightarrow \text{budziszyn}(x)) \land \ldots) ...$$

After the second phase of the integration procedure is finished and the final knowledge tree $T_K$ has been computed, the background knowledge axioms are generated from $T_K$. The resulting axioms are
Fig. 13. Concepts from WordNet

Fig. 14. Knowledge graph $G_Y$ with results of two queries to YAGO

Fig. 15. Fragment of knowledge tree $T_K$ after integration of results from YAGO
added into the FOLE formulas of the input RTE problem. Such an extended input problem is passed over to the inference process (see Fig. 8) and solved correspondingly. For further details see [Wot10] or [HWC11].

In Fig. 16 the extracted YAGO concepts are shown for the example. In Fig. 17 the knowledge tree after processing the concepts from WordNet and YAGO are shown.

![Concepts from YAGO (T and H)](image)

**Fig. 16. Concepts from YAGO (T and H)**

## 4 Conclusions and Further Developments

For military purposes it is necessary to analyze large quantities of intelligence reports and other documents written in different languages. The question is how we can automate the content extraction of these documents. In this paper we described the approach we pursued in the mIE project (“Multilingual content analysis with semantic inference on military relevant texts”). The content extraction in the mIE system is based on a combination of deep and shallow parsing with logical inferences on the analysis results and background knowledge. We briefly contrasted the ZENON project to the mIE project. In the main part of the paper, the mIE project was presented. After explaining the combined deep and shallow parsing approach with Head-driven Phrase Structured Grammars, the inference process was introduced. Then, we show how background knowledge (WordNet, YAGO) was integrated into the logical inferences to increase the accuracy of the content extraction. The prototype was also presented.

There are a lot of possibilities to further increase the capabilities of the mIE system:

- The Arabic HPSG grammar is only a very small one. Extending this grammar would also extend the capability of the content extraction from Arabic texts.
- During the inference process only the most probable meaning of the words is considered. Considering as well other - less probable - meanings might increase the inferential power.
- Because of a huge coverage of YAGO, it was almost always possible, to find information we needed for the proof. Nevertheless, it would be interesting to look at the inconsistent cases of the inference process. They were caused by errors in presupposition and anaphora resolution, incorrect syntactic derivations, and inadequate semantic representations.
During the access to YAGO at the moment only ontological relations like, e.g., type, subClassOf, or isCalled are processed. For the implementation of some temporal calculus, also temporal relations such as during, since, or until could be considered.

Other external background knowledge might be integrated, e.g., OpenCyc [MCWD06] or DBpedia [ABK+07].

References


