Recognizing Textual Entailment with Deep-Shallow Semantic Analysis and Logical Inference

Andreas Wotzlaw and Ravi Coote
Fraunhofer Institute for Communicaton, Information Processing and Ergonomics FKIE
Neuenahrer Str. 20, 53343 Wachtberg, Germany
Email: {andreas.wotzlaw, ravi.coote}@fkie.fraunhofer.de

***Abstract*—In this paper, the architecture and evaluation of a new system for recognizing textual entailment (RTE) is presented. It is conceived as an adaptable and modular environment allowing for a high-coverage syntactic and semantic text analysis combined with logical inference. For the syntactic and semantic analysis it combines an HPSG-based deep semantic analysis with a shallow one supported by statistical models in order to increase the quality and accuracy of results. For recognizing textual entailment we use logical inference of first-order employing model-theoretic techniques and automated reasoning tools. The inference is supported with problem-relevant background knowledge extracted automatically and on demand from external sources like, e.g., WordNet, YAGO, and OpenCyc, or other, experimental sources with, e.g., manually defined presupposition resolutions, or with general and common sense knowledge. The system comes with a graphical user interface for control and presentation purposes. The evaluation shows that the success rate of the presented RTE system is comparable with that of the best logic-based approaches.**

***Keywords*-recognizing textual entailment; semantic analysis; logical inference; knowledge integration; semantic reasoning.**

I. INTRODUCTION

In this paper, we present a new system for *recognizing textual entailment* (RTE, see [1], [2]). Our aim is to provide a robust, modular, and highly adaptable environment for a linguistically motivated large-scale semantic text analysis. In RTE we want to identify automatically the type of a logical relation between two input texts. In particular, we are interested in proving the existence of an entailment between them. The concept of *textual entailment* indicates the situation 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 [3]. RTE is without doubt one of the ultimate challenges for any natural language processing (NLP) system. If it succeeds with reasonable accuracy, it is a clear indication for some thorough understanding how language works. As a generic problem, it has many useful applications in NLP [4]. Interestingly, many application settings like, e.g., information retrieval, paraphrase acquisition, question answering, or machine translation can fully or partly be modeled as RTE [2]. 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., during the RTE Challenges [2].

In our setting, we try to recognize the type of the logical relation between two English 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 [5]:

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

We aim to solve an RTE problem by applying a *model-theoretic* approach where a formal *semantic representation* of the RTE problem, i.e., of the texts T and H, is computed. However, in contrast to *automated deduction* systems [6], 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 semantic representations for input problems, we build on a combination of deep and shallow techniques for semantic analysis. The main problem with approaches processing the text in a shallow fashion is that they can be tricked easily, e.g., by negation, or by 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 (e.g., short text message). 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 [7]. 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 [8]). However, for their further usage in the logical inference, the MRS expressions

ISBN: 978-1-61208-104-5

are translated into another, semantic equivalent representation of *First-Order Logic with Equality* (FOLE) [5]. This logical form with a well-defined model-theoretic semantics was already successfully applied for RTE in [9].

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 [3], [10]. The logical inference in our system 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., [2], [9]), our system enables flexible integration of background knowledge from more than one external source (see Section IV-A for details). In its current implementation, our system supports RTE, but can also be used for other NLP tasks like, e.g., large-scale syntactic and semantic analysis of English texts, or multilingual information extraction.

In the remainder of the paper, we give first a short overview of related work (Section II). Then we present in detail the architecture of our system (Section III) and explain how its success rate can be improved by employing external knowledge and presupposition resolvers (Section IV). The paper concludes with a discussion of the results (Section V).

II. Related work

Our work was inspired by the ideas given in [5], [9], where a similar, model-theoretic approach was used for the semantic text analysis with logical inference. However, in contrast to our MRS-based approach, they apply *Discourse Representation Theory* [11] for the computation of full semantic representations. Furthermore, we use the framework *Heart of Gold* [7] as a basis for the semantic analysis. For a good overview of a combined application of deep and shallow NLP methods for RTE, we refer to [7], [12]. The application of logical inference techniques for RTE was already elaborately presented in [10], [13], [14]. A discussion on formal methods for the analysis of the meaning of natural language expressions can be found in [15].

III. System Architecture

Our system for RTE provides the user with a number of essential functionalities for syntactic, semantic, and logical textual analysis, which can selectively be overridden or specialized in order to provide new or more specific ones, e.g., for anaphora resolution or word sense disambiguation. In its initial form, the application supplies, among other things, flexible program interfaces and transformation components, allows for execution of a deep-shallow syntactic and semantic analysis, integrates external inference machines and background knowledge, maintains the semantic analysis and the inference process, and provides the user with a graphical interface for control and presentation purposes.

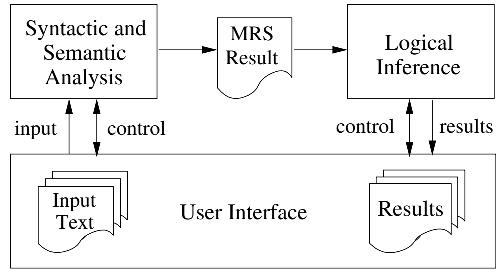

Figure 1. Overall architecture of the system.

In the following, we describe our system for RTE in more detail. It consists of three main modules (see Figure 1):

1) *Syntactic and Semantic Analysis*, where the combined deep-shallow semantic analysis of the input text is performed;
2) *Logical Inference*, where the logical inference process is implemented, supported by 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.

In the rest of the section, we discuss the way the particular modules of the system work. To make our description as comprehensible as possible, we make use of a small RTE problem. With its help we explain some crucial aspects of that how our system proceeds while trying to solve RTE problems. More specifically, we want to identify the logical relation between text T:

> *London's Tower Bridge is one of the most recognizable bridges in the world. Many falcons inhabit its old roof nowadays.*

and hypothesis H:

> *Birds live in London.*

To prove this textual entailment automatically, among other things, a precise semantic representation of the problem must be computed, the anaphoric reference between *Tower Bridge* and *its* in T must be resolved, and world knowledge (e.g., that *Tower Bridge* is in *London*) as well as ontological relations between the concepts (e.g., that *falcons* are *birds*) must be provided to the logical inference. We show how our system works while solving problems of such complexity.

A. Syntactic and Semantic Analysis

The texts of the input RTE problem after entering the system via the user interface 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 Figure 2). 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

 ISBN: 978-1-61208-104-5

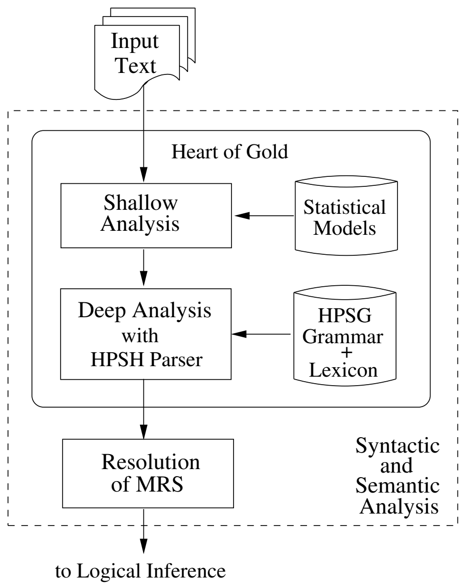

Figure 2. Module for syntactic and semantic analysis.

strategies. Here, we use its slightly modified standard configuration for English centered around the English Resource HPSG Grammar (ERG, see [16]). 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 with the Java tool JTok, the part-of-speech tagging with the statistical tagger TnT [17] trained for English on the Penn Treebank [18], and the named entity recognition with SProUT [19]. 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 [20]. 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 the *Robust Minimal Recursion Semantics* (RMRS, see [21]), when a given NLP component supports it natively. 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. Thus, RMRS is well suited for representing output also from shallow NLP components. This can be seen as a clear advantage over approaches based strictly on some specified semantic representation like those presented, e.g., in [13], [22].

Furthermore, RMRS is a common semantic formalism for HPSG grammars within the context of the *LinGO Grammar Matrix* [23]. Besides ERG, which we use for English, there are also grammars for other languages like, e.g., the Japanese HPSG grammar *JaCY* [24], the *Korean Resource Grammar* [25], the *Spanish Resource Grammar* [26], or the proprietary German HPSG grammar [27]. 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 achieved. To our best knowledge, it would be the first time that RTE problems in languages other than English could be considered.

The combined results of the deep-shallow analysis in RMRS form are transformed into MRS and resolved with Utool 3.1 [28]. Utool translates the input first from MRS into dominance constraints [29], a closely related scope underspecification formalism, and then enumerates in polynomial time all text readings represented by the dominance graph. In the current implementation, one of the most reasonable readings is chosen manually by the analyst for the further processing. A full automation of this task is still not possible in the current state-of-the-art. It requires much more knowledge about the RTE problem itself and about the discourse background. This important problem will be part of the further investigations.

For our small RTE example, the result of the combined syntactic and semantic analysis for H in form of RMRS, given as attribute value matrix, is presented in Figure 3. The results of the shallow analysis (marked bold) describe the named entities from H. Subsequently, the structure is transformed into MRS and resolved by Utool. The resulting first-order MRS in Prolog notation for the hypothesis H from our example is given below. The predicates with `_q_`, `_n_`, `_v_`, and `_p_` in their names represent quantifiers, nouns, verbs, and prepositions, respectively.

```
udef_q_rel(X6,
    bird_n_1_rel(X6),
    proper_q_rel( X9, and(
        named_rel(X9, london), and(
        locname_rel(london, X9),
        loctype_rel(city, X9))), and(
        live_v_1_rel(E2, X6),
        in_p_dir_rel(E10, E2, X9))))).
```

B. Logical Inference

The results of the semantic analysis in form of specified MRS combining deep-shallow predicates are translated into another, logical equivalent semantic representation FOLE (see Figure 4). The rule-based transformation conveys argument structure with a neo-Davidsonian analysis with semantic roles [30]. A definite article is translated according to the theory of definite description of Russell [31]. Temporal relations are modeled by adding additional predicates similar to [9], i.e., without explicit usage of time operators. Furthermore, it is possible to extend the translation mechanism to cover plural and modal forms. Appropriate ideas can be found, e.g., in [9], [32]. However, by applying them, one

 ISBN: 978-1-61208-104-5

$$\begin{bmatrix} \text{TEXT} & \\ \text{TOP} & h1 \\ \text{RELS} & \left\{ \begin{array}{l} \begin{bmatrix} \textbf{locname\_rel} & \\ \text{LBL} & h19000108 \\ \text{ARG0} & x19000108 \\ \text{CARG} & \text{london} \\ \text{ARG1} & x9 \end{bmatrix} \begin{bmatrix} \textbf{loctype\_rel} & \\ \text{LBL} & h19000110 \\ \text{ARG0} & x19000110 \\ \text{CARG} & \text{city} \\ \text{ARG1} & x9 \end{bmatrix} \begin{bmatrix} \textit{udef\_q\_rel} & \\ \text{LBL} & h3 \\ \text{ARG0} & x6 \\ \text{RSTR} & h5 \\ \text{BODY} & h4 \end{bmatrix} \begin{bmatrix} \textit{\_bird\_n} & \\ \text{LBL} & h7 \\ \text{ARG0} & x6 \end{bmatrix} \\ \begin{bmatrix} \textit{\_live\_v} & \\ \text{LBL} & h8 \\ \text{ARG0} & e2 \\ \text{ARG1} & x6 \end{bmatrix} \begin{bmatrix} \textit{\_in\_p} & \\ \text{LBL} & h10001 \\ \text{ARG0} & e10 \\ \text{ARG1} & e2 \\ \text{ARG2} & x9 \end{bmatrix} \begin{bmatrix} \textit{proper\_q\_rel} & \\ \text{LBL} & h11 \\ \text{ARG0} & x9 \\ \text{RSTR} & h13 \\ \text{BODY} & h12 \end{bmatrix} \begin{bmatrix} \textit{named\_rel} & \\ \text{LBL} & h14 \\ \text{ARG0} & x9 \\ \text{CARG} & \text{London} \end{bmatrix} \end{array} \right\} \\ \text{HCONS} & \{h5 \text{ qeq } h7, h13 \text{ qeq } h14\} \\ \text{ING} & \{h8 \text{ ing } h10001\} \end{bmatrix}$$

Figure 3. RMRS as attribute value matrix for hypothesis H from the example.

needs to be careful since the complexity and the amount of the resulting FOLE formulas will grow rapidly, making the input problem apparently much harder to solve.

The translated FOLE formulas are stored locally and can be used for the further analysis. Furthermore, such formally expressed input text can and *should* be extended with additional knowledge in form of *background knowledge axioms*. The additional axioms are formulated in FOLE and integrated into the input problem. The integration of background knowledge will be discussed in detail in Section IV.

As an example here, the translation of the specified MRS into FOLE for the hypothesis H from our example given earlier in Section III-A produces the following formula with a neo-Davidsonian event representation:

```
some(X6,and(
    bird_n_1(X6),
    some(X9,and(and(
        named_r_1(X9),and(
        location_n_1(X9),and(
        london_loc_1(X9),
        city_n_1(X9)))),
        some(E2,and(
            event_n_1(E2),and(and(
            live_v_1(E2),
            agent_r_1(E2,X6)),
            in_r_1(E2,X9)))))))).
```

C. Inference Process

The goal here is to prove the logical relation between two input texts represented formally by corresponding FOLE formulas. We are interested in answering the question whether the relation is an entailment, a contradiction, or whether maybe the hypothesis H provides just new information with respect to the text T (i.e., is informative, see Section I). To check which type of a logical relation for the input problem holds, we use two kinds of automated reasoning tools:

- *Finite model builders*: Mace 2.2 [33], Paradox 3.0 [34], and Mace4 [35], and
- *First-order provers*: Bliksem 1.12 [36], Otter 3.3 [37], Vampire 8.1 [38], and Prover9 [39].

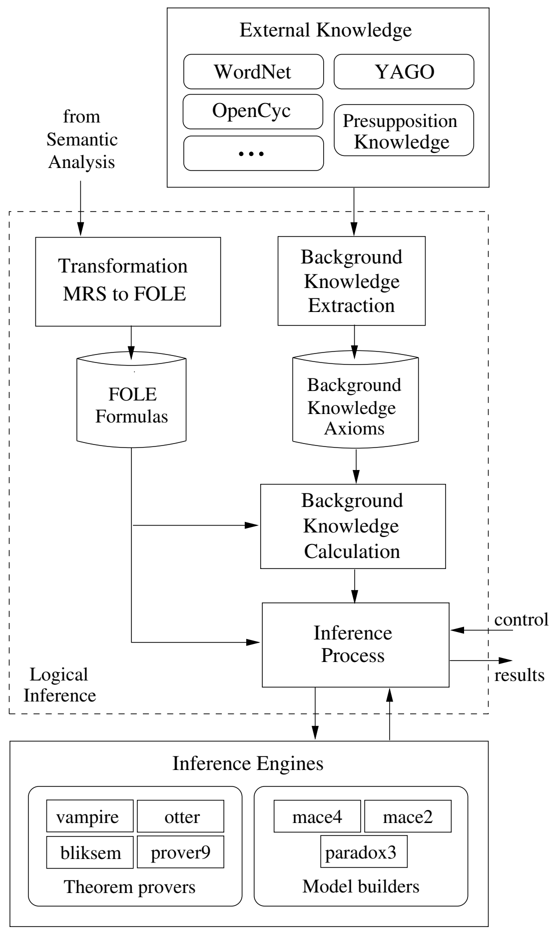

Figure 4. Module for logical inference with external inference machines and background knowledge.

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 [40]. Model builders are designed to show that a formula is true in at least one model. The experiments with

ISBN: 978-1-61208-104-5

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 [9], [40], 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.

All reasoning machines were developed to deal with inference problems stated in FOLE. They are successfully integrated into our system for RTE. To this end, we use a translation from FOLE into the formats required by the inference tools. Furthermore, the user can specify via the user interface which inference machines (i.e., which theorem prover and which model builder) should be used by the inference process. The tests have shown that the efficiency and the success of solving a given RTE problem depend much on the inference machines chosen for it.

D. User Interface

The results of the syntactic processing, semantic construction, and logical inference like, e.g., HPSG and MRS structures, FOLE formulas, models, proofs, integrated background knowledge, and other detailed information are presented to the user within a dedicated GUI. With its help, one can further customize and control both the semantic and logical analysis, e.g., choose the input text or the background knowledge source, inspect the results of shallow-deep analysis, or select other inference machines.

IV. Improving the inference quality

Many applications in modern information technology utilize ontological background knowledge. This applies particularly to the applications from the Semantic Web, but also to other domains like, e.g., information retrieval, question answering, or recognizing textual entailment. The existing RTE applications today use typically only one source of background knowledge, e.g., WordNet [41] or Wikipedia. However, they could boost their performance if a huge ontology with knowledge from several sources were available. We show here how more than one knowledge source can be used successfully for RTE. In this paper, we mean by ontology any set of facts and/or axioms comprising potentially both individuals (e.g., London) and concepts (e.g., city).

The inference process needs background knowledge to support its proofs. However, with increasing number of background knowledge axioms the search for finite models becomes more time-consuming. Thus, only problem-relevant knowledge should be considered in the inference process.

A. Background Knowledge

Our RTE system supports the extraction of background knowledge from different kinds of sources (see Figure 4). It supplies problem-relevant background knowledge automatically as first-order axioms and integrates them into the input RTE problem. WordNet 3.0 is used as lexical knowledge source for synonymy, hyperonymy, and hyponymy relations. With WordNet we try to detect entailments between lexical units from the text and the hypothesis. Axioms of generic knowledge cover the semantics of possessives, active-passive alternation, and spatial knowledge (e.g., that Tower Bridge is located in London). YAGO [42] with facts automatically extracted from Wikipedia and unified with WordNet is used as a source of ontological knowledge. OpenCyc 2.0 [43] can also be used as a background knowledge source. The computation of axioms for a given problem is solved using a variant of Lesk's WSD algorithm [44].

In the following, we describe the idea we use to combine individuals and concepts from WordNet with those from YAGO in order to support RTE. Our integration technique is composed of two steps. After the first-order representation of the problem is computed and subsequently translated into FOLE, the search for relevant background knowledge begins. First, we list all predicates (i.e., concepts and individuals) from the FOLE formulas which can be used for the search. In the current implementation, we consider as *search predicates* S all nouns, verbs, and named entities, together with their sense information (i.e., their readings) specified by the last number in the predicate name, e.g., `bird_n_1`. Having the search predicates, we try to find them in WordNet and, by employing the hyperonymy/hyponymy relation, we build a *knowledge tree* T_K with leaves represented by the concepts from the formulas, whereas inner nodes and the root are coming from WordNet.

In Figure 5, we show a fragment of a knowledge tree for $\{T, H\}$ of our RTE problem from the beginning of Section III. Here, each node represents at least one concept or individual, whereas the directed edges correspond to the hyponym relations between them, e.g., the named entity `london` 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 `london`. Figure 5 depicts also one complex node representing synonymous concepts `live` and `inhabit`.

It is crucial for the integration that the sense information computed for the concepts and individuals during the semantic analysis matches exactly the senses used by external knowledge sources. This ensures that the semantic consistency is preserved across the semantic and logical analysis. However, this constitutes an extremely difficult task which does not seem to be solved fully automatically yet by any word sense disambiguation technique. Since in WordNet but also in ERG the senses are ordered from most to least frequently used, with the most common sense numbered `1`, we take in the current implementation for semantic representations generated during the semantic analysis the most frequent concepts from ERG.

 ISBN: 978-1-61208-104-5

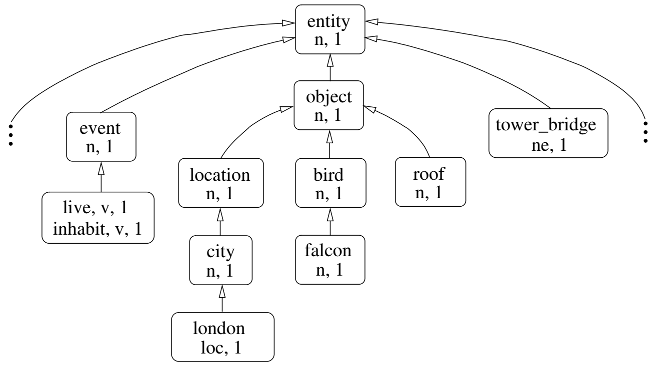

Figure 5. Example of knowledge tree for RTE. Here, `v`, `n`, `loc`, and `ne` stand for verb, noun, location, and named entity, respectively, whereas the numbers represent the sense information.

In the second step of our integration technique, we consult YAGO about the predicates from S that were not found in WordNet during the first step. If succeed, YAGO returns a directed acyclic graph (DAG) G_K with new concepts which classify those concepts that were not recognized before. Unfortunately, as a DAG, it cannot be integrated completely into the knowledge tree T_K. Our experiments have shown that a knowledge graph, when represented as a tree, assures that the set of background knowledge axioms, which will be generated afterwards from that tree stays consistent (i.e., it includes no contradictions). Thus, in order to preserve the consistency and correctness of the results, we select for the integration into the knowledge tree T_K only those concepts and relations from G_K, which lay on the longest path from the root to one of its leaves and which has the most common nodes with the knowledge tree T_K from the first step. This heuristic can cause some loss of effectivity of the entire RTE inference process, since some concepts which are relevant for the RTE problem might not be integrated as background knowledge into it. Nevertheless, because of its acceptable performance while solving problems from the development sets of the past RTE Challenge [4], we have decided to use it as a good starting point for the further research.

After the background knowledge tree T_K has been extended, the knowledge axioms are generated from it. We generate axioms expressing the hyperonymy/hyponymy relations (i.e., ontological relations is-a and is-not-a) and the synonymy relations (is-eq) in T_K. For the knowledge tree given in Figure 5, the following axioms (here not a complete list) can be generated.

$$\forall x(city\_n\_1(x) \rightarrow location\_n\_1(x))$$
$$\forall x(event\_n\_1(x) \rightarrow \neg object\_n\_1(x))$$
$$\forall x(live\_v\_1(x) \leftrightarrow inhabit\_v\_1(x))$$

B. Presupposition resolution

Many words and phrases trigger presuppositions which have clearly semantic content important for the inference process. We try to represent some of them explicitly. Our trigger-based mechanism uses noun phrases as triggers, but it can be extended to verb phrases, particles, etc. After a presupposition is triggered, the mechanism resolves it, and integrates it as a new FOLE axiom into the RTE problem. The automatic axiom generation is based on λ-conversion and employs *abstract axioms* and a set with possible *axiom arguments*. The axioms and their arguments are still part of an experimental knowledge source (see Presupposition Knowledge in Figure 4). Here is an example for an abstract axiom which allows for a translation from a noun phrase into an intransitive verb phrase:

$$\lambda P[\lambda R[\lambda S[\forall x_1(\forall x_2(P@x_1 \wedge R@x_2 \wedge nn\_r\_1(x_1, x_2) \rightarrow \exists x_3(R@x_3 \wedge \exists x_4(S@x_4 \wedge event\_n\_1(x_4) \wedge agent\_r\_1(x_4, x_3))))))]]]. \quad (1)$$

If text T (expressed with FOLE formulas) contains a noun phrase being a key for some entry in the set of possible axiom arguments, then the arguments pointed by that key are applied to their abstract axiom, and a new background axiom is generated. For a complex noun phrase *price explosion* with its semantic representation $price\_explosion\_n\_1$, the following arguments can be considered:

$$\lambda x[explosion\_n\_1(x)],\ \lambda x[price\_n\_1(x)],\ \text{and}\ \lambda x[explode\_v\_1(x)],$$

which after being applied to the abstract axiom (1) produce the following background knowledge axiom:

$$\forall x_1(\forall x_2(explosion\_n\_1(x_1) \wedge price\_n\_1(x_2) \wedge nn\_r\_1(x_1, x_2) \rightarrow \exists x_3(price\_n\_1(x_3) \wedge \exists x_4(explode\_v\_1(x_4) \wedge event\_n\_1(x_4) \wedge agent\_r\_1(x_4, x_3))))). \quad (2)$$

The presupposition axioms having complexity similar to (2) are first combined with the existing background knowledge axioms and finally integrated as background knowledge into the input RTE problem.

V. Conclusion and Future Work

In this paper, a new adaptable, linguistically motivated system for RTE was presented. Its deep-shallow semantic analysis, employing a broad-coverage HPSG grammar ERG, was combined with a logical inference process supported by an extended usage of external background knowledge. The architecture of the system was given in detail and its functionality was explained with several examples.

The system was successfully implemented and evaluated in terms of success rate and efficiency. For now, it is still impossible to measure its exact semantic accuracy as there is no corpus with gold standard representations which would make comparison possible. Measuring semantic adequacy could be done systematically by running the system on controlled inference tasks for selected semantic phenomena.

For our tests, we used the RTE problems from the development set of the third RTE Challenge [4]. Our system with was able to solve correctly 64 percent of the RTE problems. This is better than the most of the other approaches from that RTE Challenge which are based on some deep approach combined with logical inference. Unfortunately, it is still not as good as the success rate of 72 percent obtained by the best logic-based semantic approach given by [14]. This can be explained, among other things, by a more extensive and fine-grained usage of specific semantic phenomena, e.g., a sophisticated analysis of named entities, in particular person names, distinguishing first names from last names. This shows, however, that extending our system with similar techniques for more accurate treatment of specific semantic phenomena should further improve its success rate.

Nevertheless, it is interesting to look at the inconsistent cases of the inference process which were produced during the evaluation. They were caused by errors in presupposition and anaphora resolution, incorrect syntactic derivations, and inadequate semantic representations. They give us good indications for further improvements. Here, particularly the word sense disambiguation problem will play a decisive role for matching the set of senses of the semantic analyzers with multiple, and likely different, sets of senses from the different knowledge resources. Once tackled more precisely, it should decisively improve the success rate of the system.

As being still work-in-progress, we plan to extend our system with methods for word sense disambiguation, paraphrase detection, and a better anaphora resolution within a discourse. We consider also enhancing the logical inference module with statistical inference techniques in order to improve its performance and recall. Since the strength but in some respects also the weakness of our system lies in the difficulties regarding the computation of a full semantic representation of the input problem (see, e.g., [45] for a good discussion), it might be recommended to integrate into the system some models of natural language inference which identifies valid entailments by their lexical and syntactic features, without full semantic interpretation like, e.g., the one proposed by MacCartney and Manning [46].

ACKNOWLEDGMENT

The authors would like to thank Matthias Hecking for many fruitful discussions and comments on the work presented in this paper.

REFERENCES

[1] I. Dagan, B. Dolan, B. Magnini, and D. Roth, "Recognizing textual entailment: Rational, evaluation and approaches," *Natural Language Engineering. Special Issue on Textual Entailment*, vol. 15, no. 4, pp. i–xvii, 2009.

[2] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini, "The fifth PASCAL recognizing textual entailment challenge," in *TAC 2009 Workshop*, Gaithersburg, Maryland, 2009.

[3] J. Bos, "Towards wide-coverage semantic interpretation," in *Proceedings of the 6th International Workshop on Computational Semantics IWCS-6*, 2005, pp. 42–53.

[4] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The third PASCAL recognizing textual entailment challenge," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, 2007, pp. 1–9.

[5] P. Blackburn and J. Bos, *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, 2005.

[6] E. Akhmatova, "Textual entailment resolution via atomic propositions," in *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, UK, 2005, pp. 61–64.

[7] U. Schäfer, "Integrating deep and shallow natural language processing components – representations and hybrid architectures," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2007.

[8] A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag, "Minimal recursion semantics: An introduction," *Research on Language and Computation*, vol. 3, pp. 281–332, 2005.

[9] J. R. Curran, S. Clark, and J. Bos, "Linguistically motivated large-scale NLP with C&C and boxer," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, 2007, pp. 33–36.

[10] J. Bos and K. Markert, "When logical inference helps determining textual entailment (and when it doesn't)," in *Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment*, Venice, Italy, 2006.

[11] H. Kamp and U. Reyle, *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers, 1993.

[12] J. Bos and K. Markert, "Combining shallow and deep NLP methods for recognizing textual entailment," in *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, UK, 2005, pp. 65–68.

[13] ——, "Recognising textual entailment with logical inference," in *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vancouver, Canada, 2005, pp. 628–635.

[14] M. Tatu and D. Moldovan, "A logic-based semantic approach to recognizing textual entailment." in *Proceedings of the COLING/ACL on Main conference poster sessions*, Morristown, NJ, 2006, pp. 819–826.

[15] J. Bos, "Let's not argue about semantics," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008, pp. 28–30.

[16] D. Flickinger, "On building a more efficient grammar by exploiting types," *Natural Language Engineering*, vol. 6, no. 1, pp. 15–28, 2000.

[17] T. Brants, "TnT – a statistical part-of-speech tagger," in *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA, 2000, pp. 224–231.

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[19] W. Drożdżyński, H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu, "Shallow processing with unification and typed feature structures – foundations and applications," *Künstliche Intelligenz*, vol. 18, no. 1, pp. 17–23, 2004.

[20] U. Callmeier, "PET – a platform for experimentation with efficient HPSG processing techniques," *Natural Language Engineering*, vol. 6, no. 1, pp. 99–108, 2000.

[21] A. Copestake, "Report on the design of RMRS," University of Cambridge, UK, Tech. Rep. D1.1b, 2003.

[22] P. Blackburn, J. Bos, M. Kohlhase, and H. D. Nivelle, "Automated theorem proving for natural language understanding," in *Problemsolving Methodologies with Automated Deduction (Workshop at CADE-15)*, 1998.

[23] E. M. Bender, D. Flickinger, and S. Oepen, "The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars," in *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 2002.

[24] M. Siegel and E. M. Bender, "Efficient deep processing of japanese," in *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization. Coling 2002 Post-Conference Workshop*, 2002.

[25] K. Jong-Bok and Y. Jaehyung, "Parsing mixed constructions in a typed feature structure grammar," *Lecture Notes in Artificial Intelligence*, vol. 3248, pp. 42–51, 2005.

[26] M. Marimon, "Integrating shallow linguistic processing into a unification-based spanish grammar," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, 2002.

[27] B. Cramer and Y. Zhang, "Construction of a German HPSG grammar from a detailed treebank," in *Proceedings of the Workshop on Grammar Engineering Across Frameworks*. Association for Computational Linguistics, 2009.

[28] A. Koller and S. Thater, "Efficient solving and exploration of scope ambiguities," in *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, Ann Arbor, Michigan, 2005, pp. 9–12.

[29] S. Thater, "Minimal recursion semantics as dominance constraints: Graph-theoretic foundation and application to grammar engineering," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2007.

[30] D. Dowty, "On semantic content of the notion of "thematic role"," in *Properties, Types and Meaning*, G. C. Barbara Partee and R. Turner, Eds. Dordrecht (Kluwer), 1989, vol. 2, pp. 69–129.

[31] B. Russell, "On denoting," *Mind, New Series*, vol. 14, no. 56, pp. 479–493, 1905.

[32] H. Lohnstein, *Formale Semantik und natürliche Sprache. Einführendes Lehrbuch.* Westdeutscher Verlag, 1996.

[33] W. McCune, *Mace 2.0 Reference Manual and Guide*, Argonne National Laboratory, IL, 2001.

[34] K. Claessen and N. Sörensson, "New techniques that improve MACE-style model finding," in *Proceedings of the CADE-19 Workshop: Model Computation – Principles, Algorithms, Applications*, Miami, FL, 2003.

[35] W. McCune, *Mace4 Reference Manual and Guide*, Argonne National Laboratory, IL, 2003.

[36] H. de Nivelle, "Bliksem 1.10 user manual," URL: http://www.ii.uni.wroc.pl/˜nivelle/software/bliksem, 2003.

[37] W. McCune, *OTTER 3.3 Reference Manual*, Tech. Memo ANL/MCS-TM-263, Argonne National Laboratory, Argonne,IL, Argonne National Laboratory, IL, 2003.

[38] A. Riazanov and A. Voronkov, "The design and implementation of VAMPIRE," *AI Commun.*, vol. 15, no. 2,3, pp. 91–110, 2002.

[39] W. McCune, "Prover9 manual," URL: http://www.cs.unm.edu/˜mccune/prover9/manual/2009-11A/, Argonne National Laboratory, Argonne, IL, 2009.

[40] J. Bos, "Exploring model building for natural language understanding," in *Proceedings of ICoS-4*, 2003, pp. 25–26.

[41] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[42] F. Suchanek, G. Kasneci, and G. Weikum, "YAGO - a large ontology from Wikipedia and WordNet," *Elsevier Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[43] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira., "An introduction to the syntax and content of Cyc," in *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Stanford, CA, 2006.

[44] S. Banerjee and T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet," in *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, London, UK, 2002, pp. 136–145.

[45] A. Burchardt, N. Reiter, S. Thater, and A. Frank, "A semantic approach to textual entailment: System evaluation and task analysis," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[46] B. MacCartney and C. D. Manning, "An extended model of natural logic," in *Proceedings of the 8th International Conference on Computational Semantics (IWCS-8)*, 2009, pp. 140–156.

 ISBN: 978-1-61208-104-5