IJCLA VOL. 1, NO. 1-2, JAN-DEC 2010, PP. 105-119
RECEIVED 26/11/09 ACCEPTED 16/01/10 FINAL 12/03/10

Hypernymy Extraction Using a Semantic Network Representation

TIM VOR DER BRÜCK

IICS, Germany

ABSTRACT

There are several approaches to detect hypernymy relations from texts by text mining. Usually these approaches are based on supervised learning and in a first step are extracting several patterns. These patterns are then applied to previously unseen texts and used to recognize hypernym/hyponym pairs. Normally these approaches are only based on a surface representation or a syntactical tree structure, i.e., constituency or dependency trees derived by a syntactical parser. In this work, however, we present an approach that operates directly on a semantic network (SN), which is generated by a deep syntactico-semantic analysis. Hyponym/hypernym pairs are then extracted by the application of graph matching. This algorithm is combined with a shallow approach enriched with semantic information.

1 INTRODUCTION

Quite a lot of work has been done on hypernymy extraction in natural language texts. The approaches can be divided into three different types of methods:
- Analyzing the syntagmatic relations in a sentence
- Analyzing the paradigmatic relations in a sentence
- Document clustering

The first type of algorithms usually employ a set of patterns. Quite popular patterns were proposed by Hearst, the so–called Hearst patterns [1]. The following Hearst patterns are defined:

- NP_{hyper} such as $\{\{NP_{hypo},\}$* (and|or)$\}$ NP_{hypo}
- such NP_{hyper} as $\{NP_{hypo},\}$* $\{$(and|or)$\}$ NP_{hypo}
- NP_{hypo} $\{$,$NP_{hypo}\}$*$\{$,$\}$ or other NP_{hyper}
- NP_{hypo} $\{$,$NP_{hypo}\}$*$\{$,$\}$ and other NP_{hyper}
- $NP_{hyper}\{,\}$ including $\{NP_{hypo},\}$* $\{$and|or$\}$ NP_{hypo}
- $NP_{hyper}\{,\}$ especially $\{NP_{hypo},\}$* $\{$and|or$\}$ NP_{hypo}

These patterns are applied on arbitrary texts and the instantiated variables NP_{hypo} and NP_{hyper} are then extracted as a concrete hypernymy relation. Several approaches were developed to extract such patterns automatically from a text corpus by either employing a surface representation [2] or a syntactical tree structure [3].

Instead of applying the patterns to an ordinary text corpus, some approaches apply them on the entire Internet by transferring the patterns into Web search engine queries[4, 5]. Pattern learning and application is combined by the system KnowItAll [6] which uses a bootstrapping mechanism to extend patterns and extracted relations iteratively. An alternative approach to pattern matching is the usage of kernel functions where the kernel function defines a similarity measure between two syntactical trees possibly containing a hypernymy or an other semantic relation[7].

Paradigmatic approaches expect that words in the textual context of the hypernym (e.g., neighboring words) can also occur in the context of the hyponym. The textual context can be represented by the set of the words which frequently occur together with the hypernym (or the hyponym). Whether a word is the hypernym of a second word can then be determined by a similarity measure on the two sets [5].

A further often employed method for extracting hypernyms is document clustering. For that, the documents are hierarchically clustered. Each document is assigned a concept or word it describes. The document hierarchy is then transferred to a concept or word hierarchy[8].

In contrast to the formerly mentioned methods, we will follow a purely semantic approach to extract hypernymy relations between concepts (word readings) instead of words which operates on semantic networks (SN) rather than on syntactical trees or surface representations. By using a semantic representation, the patterns are more generally applicable and therefore the number of patterns can be reduced.

In the first step, the entire content of the German Wikipedia corpus is transformed into SNs following the MultiNet[1] formalism[9]. Afterwards, deep patterns are defined which are intended to be matched to that SNs.

---

[1] MultiNet is the abbreviation of **Multi**layered Extended Semantic **Net**works

Some of them are learned by text mining on the SN representations, some of them are manually defined.

After the patterns are applied on the Wikipedia corpus, the ontological sorts and features of the extracted hyponym and hypernym, as defined by the MultiNet formalism (see Sect. 2), are compared to filter out incorrect concept pairs. Finally, we determine a confidence score for all remaining relations which reflects the likelihood that the hypernymy relation has actually been correctly recognized.

This approach is combined with a shallow method based on Hearst patterns enriched with semantic information if present. The shallow patterns are defined as regular expressions and are applied on the token list which is always present independent of the fact that the SN is successfully constructed.

2 MultiNet

MultiNet is a SN formalism. In contrast to SNs like WordNet [10] or GermaNet [11], which contain lexical relations between synsets, MultiNet is designed to comprehensively represent the semantics of natural language expressions. A SN in the MultiNet formalism is given as a set of nodes and edges where the nodes represents the concepts (word readings) and the edges the relations (or functions) between the concepts. Example SNs are shown in Fig. 1 and Fig. 2. Important MultiNet relations/funtions are [9]:

- SUB: Relation of conceptual subordination (hyponymy)
- AGT: Conceptual role: Agent
- ATTR: Specification of an attribute
- VAL: Relation between a specific attribute and its value
- PROP: Relation between object and property
- *ITMS: Function enumerating a set
- PRED: Predicative concept characterizing a plurality
- OBJ: Neutral object
- SUBS: Relation of conceptual subordination (for situations)

It is differentiated between lexicalized nodes (i.e., associated to entries in the semantic lexicon) and nodes which represents complex situations or individual objects, and are not associated with single lexical entries. The latter nodes are just assigned a unique ID.

MultiNet is supported by a semantic lexicon [12] which defines, in addition to traditional grammatical entries like gender and number, one or more ontological sorts and several semantic features for each lexicon entry. The ontological sorts (currently more than 40) form a taxonomy. In contrast to other taxonomies ontological sorts are not necessarily lexicalized, i.e., they do not necessarily denote lexical entries. The following list shows a small selection of ontological sorts which are derived from *object*:

- Concrete objects: e.g., *milk*, *honey*
  - Discrete objects: e.g., *chair*
  - Substances: e.g., *milk*, *honey*
- Abstract objects: e.g., *race*, *robbery*

Semantic features denote certain semantic properties for objects. Such a property can either be present, not present or underspecified. A selection of several semantic features is given below:

- ANIMAL
- ANIMATE
- ARTIF (artificial)
- HUMAN
- SPATIAL
- THCONC (theoretical concept)

Example for the concept *house.1.1*[2]: discrete object; ANIMAL -, ANIMATE -, ARTIF +, HUMAN -, SPATIAL +, THCONC -, ...

The SNs following the MultiNet approach are constructed by the deep linguistic parser WOCADI[3][13] for German text analysis. WOCADI employs for parsing a word class functional analysis instead of a grammar.

3 Application of Deep Patterns

The employed patterns are represented as subnets of the SNs where some of the nodes are marked as slots. These slots are filled if the pattern was successfully matched to an SN. In the example depicted in Fig. 1 the hyponym can be extracted by the pattern:

$$\begin{aligned} SUB(A,B) \leftarrow SUB(C,A) \wedge PRED(E,B) \wedge \\ *ITMS(D,C,E) \wedge PROP(E, \mathit{other.1.1}) \end{aligned} \tag{1}$$

where A is instantiated to *secretary*, B to *politician* and C, B and D to non-lexicalized concepts. $*ITMS$ is a MultiNet function which combines several arguments in a conjunction. Disjunctions are combined by

[2] the suffix .1.1 denote the reading numbered .1.1 of the word house

[3] WOCADI is the abbreviation for **wo**rd **c**l**a**ss **di**sambiguation.

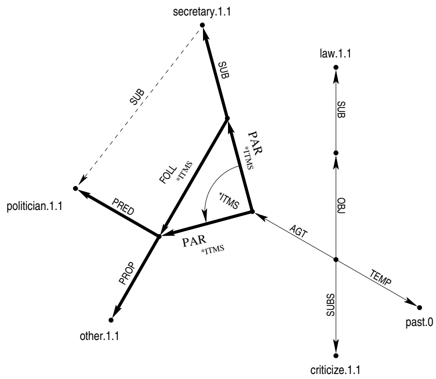

Fig. 1. Hypernymy extraction from the SN representing the sentence: *The secretary and other politicians criticized the law.* The edges matched with the pattern D_1 are printed in bold face. The edge which was inferred by the pattern is printed as a dashed line.

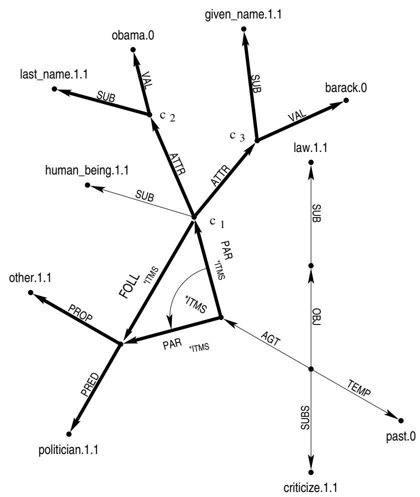

Fig. 2. Hypernymy extraction with an anthroponym in the SN representing the sentence: *Barack Obama and other politicians criticized the law.* The edges matched with the pattern D_2 are printed in bold face.

$*ALTN1/2$. However, this procedure has a serious drawback. The pattern given in Equation 1 is only applicable if the $*ITMS$ function has exactly two arguments (C,E) and one result (D). This means separate patterns are required for three and more arguments. This also implies that the patterns are rather specific, which makes learning them automatically from data difficult. Thus, we convert all functions in an SN with a variable number of arguments like $*ITMS$ and $*ALTN1/2$ to binary relations in the following way:

For each function $x_p = f(x_1, \ldots, x_n)$ with variable arguments as stated above we create n relations $PAR(x_p, x_1)$,...,$PAR(x_p, x_n)$ to represent the parent child relationships between the result and the arguments and $(n\dot{(}n-1))/2$ relations to represent the sequence of the arguments: $FOLL(x_i, x_j) \Leftrightarrow i < j$. Making the above-mentioned modifications the pattern given in Equation 1 changes to:

$$D_1 : SUB(A, B) \leftarrow SUB(C, A) \land PRED(E, B) \land \underset{*ITMS}{PAR}(D, C) \land \\ \underset{*ITMS}{PAR}(D, E) \land \underset{*ITMS}{FOLL}(C, E) \land PROP(E, \mathit{other.1.1}) \quad (2)$$

Note that different sentences can lead to the same SN. For instance, the semantically equivalent sentences *The secretary and other politicians criticized the law.* and *The secretary as well as other politicians criticized the law.* lead to the same SN, which is displayed in Fig. 1. Thus, the pattern D_1 in Equation 2 can be used to extract the relation

$$SUB(\mathit{secretary.1.1}, \mathit{politician.1.1})$$

from both sentences. In general, the number of patterns can be considerably reduced by using an SN in comparison to the employment of a shallow representation.

Furthermore, the deep semantic representation allows the simple extraction of hypernymy pairs which involve multi-token anthroponyms, like the fact that *Barack Obama* is a politician, where the extraction of multi-token names is not trivial using shallow patterns. Anthroponyms are already identified by the deep linguistic parser and represented by attribute value pairs (see Fig. 2), which allows to use a similar approach to the extraction of generic hyponyms. In contrast to generic concepts extracted anthroponyms are not stored as binary relations, but as more complex expressions:

Example:

$$\begin{aligned}N :=\ & ATTR(A,F) \wedge SUB(F, \mathit{last\_name.1.1}) \wedge VAL(F,G) \wedge \\ & ATTR(A,H) \wedge SUB(H, \mathit{given\_name.1.1}) \wedge VAL(H,I) \\ D_2 :& N \wedge SUB(A,B) \leftarrow N \wedge PRED(E,B) \wedge \underset{*ITMS}{PAR}(D,A) \wedge \\ & \underset{*ITMS}{PAR}(D,E) \wedge \underset{*ITMS}{FOLL}(A,E) \wedge PROP(E, \mathit{other.1.1})\end{aligned} \tag{3}$$

If pattern D_2 is applied on the SN shown in Fig. 2 the relations

$$\begin{aligned}& ATTR(c_1,c_2) \wedge SUB(c_2, \mathit{lastname.1.1}) \wedge VAL(c_2, \mathit{obama.0}) \wedge \\ & ATTR(c_1,c_3) \wedge SUB(c_3, \mathit{given\_name.1.1}) \wedge \\ & VAL(c_3, \mathit{barack.0}) \wedge SUB(c_1, \mathit{politician.1.1})\end{aligned} \tag{4}$$

are extracted, which denote the fact that Barack Obama is a hyponym of the concept *politician.1.1*. Note that we do not differentiate between instances (like person or country names) and hyponyms since instances and hyponyms can be extracted with almost identical patterns (especially for non-anthroponyms and shallow patterns).

4 SEMANTIC-ORIENTED FILTERING TECHNIQUES

The deep patterns described above sometimes extract concept pairs which are not related in a hypernymy relation. A two step mechanism is used to identify such concept pairs. In a first step concept pairs are filtered out if their semantic features and ontological sorts do not meet certain criteria. In the second step, several numerical features are determined for the remaining concepts and combined by the usage of a support vector machine (SVM)[14] to a confidence score. The SVM was trained on a set of annotated hypernymy relation candidates. Concept pairs assigned a high score are likely to express in fact a hypernymy relation. By using this two step approach the number of concept pairs needed to be stored in the database is reduced. In this section we will describe the filtering techniques, the scoring features are introduced in Section 5.

A hyp**o**nym is a specialization of the associated hyp**er**nym. Thus, the hyponym should have all semantic features of the hypernym (with identical values) and the ontological sort of the hyponym has to be subsumed by the sort of the hypernym (equality is allowed too).
Example: *giraffe.1.1* (animal:+) cannot be a hyponym of *house.1.1* (animal:-).

Naturally, this approach only works in all cases if the ontology is monotonic in respect to the employed semantic features. The most prominent example for non-monotonicity is the penguin. It cannot fly although its hypernym *bird.1.1* is associated the property *flying*. To account for such effects and potential misclassifications by the lexicon editor, a small mismatch is allowed.

In the MultiNet formalism, a lexical entry can be marked as a meaning molecule[9, p.292] consisting of several meaning facettes. An example is *school.1.1* which can denote either a building or an institution. If a concept is a meaning molecule, it is associated with more than one semantic feature vector and sort. In this case it is checked if there exists at least one pair of hyponym/hypernym semantic features and sorts which fulfills the above-mentioned subsumption/superset conditions.

Our semantic oriented lexicon contains more than 27 000 deep entries and more than 75 000 shallow entries. Still, in some cases, either the hyponym or hypernym candidate may not be contained which makes a check using semantic features or ontological sorts impossible. If a concept is represented by a compound noun, this problem can be solved by regarding the head instead which can be derived by a morphological analysis.

Different approaches are followed depending on whether the hypernym or the hyponym is not found in the lexicon, but the lexicon does contain its head.

If the hypernym is not contained in the lexicon, it suffices to show that its head concept C is not a hypernym of A to discharge the concept pair (A, B) of being related in a hypernymy relation which is easy to see by contradiction.

The fact that C is the head of B usually implies $SUB(B, C)$. Additionally, let us assume: $\neg\, SUB(A, C)$. Suppose $SUB(A, B)$. Then, due to the transitivity of the hypernym relation, it would follow that $SUB(A, C)$, which is known not to hold.

In the case that the potential hyponym A is not found, a different approach has to be followed. If a tree structure of the ontology is assumed then if A is a hyponym of B, the head C of A can either be a hypernym or a hyponym of B. If both of these cases can be rejected by the comparison of the ontological sorts and semantic features of C and B, the assumption that A is a hyponym of B can be rejected too. Note that theoretically, this approach could fail if the ontology is organized in a directed acyclic graph instead of a tree structure. However, no such problems were observed in practice.

5 Features Used for Scoring

We determine a confidence score for each extracted relation, which is computed by combining several numerical features described below.

Correctness Rate: The feature *Correctness Rate* takes into account that the recognized hypernym alone is already a strong indication for the correctness or incorrectness of the investigated relation. The same holds for the assumed hyponym as well. For instance, relations with hypernym *liquid* and *town* are usually recognized correctly. However, this is not the case for abstract concepts. Moreover, movie names are often extracted incompletely since they can consist of several tokens. Thus, this indicator determines how often a concept pair is classified correctly if a certain concept shows up in the first (hyponym) or second (hypernym) position. More formally, we are interested in determining the following probability:

$$p = P(h = t|\, first(rel) = a_1 \wedge sec(rel) = a_2) \tag{5}$$

where

- $first(rel)$ denotes the first concept (the assumed hyponym) in the relation rel
- $sec(ond)(rel)$ denotes the second concept (the assumed hypernym) in the relation rel
- $h(ypernym) = t(rue)$ denotes that a hypernym relation holds

Applying Bayes' theorem to Equation 5 leads to the Equation:

$$p = P(h = t) \cdot \frac{P(first(rel) = a_1 \wedge sec(rel) = a_2|h = t)}{P(first(rel) = a_1 \wedge sec(rel) = a_2)} \tag{6}$$

For better generalization, we make the assumption that the events $first(rel)$ and $sec(rel)$ as well as $(first(rel)|h = t)$ and $(sec(rel)|h = t)$ are independent. Using these assumptions, Equation 6 can be rewritten:

$$p \approx p' = P(h = t) \cdot \frac{P(\mathit{first}(rel) = a_1|h = t)}{P(first(rel) = a_1)} \cdot \frac{P(sec(rel) = a_2|h = t)}{P(sec(rel) = a_2)}$$

$$p' = \frac{P(\mathit{first}(rel) = a_1 \wedge h = t)}{P(\mathit{first}(rel) = a_1)} \cdot \frac{P(sec(rel) = a_2 \wedge h = t)}{P(h = t) \cdot P(sec(rel) = a_2)}$$

$$p' = \frac{1}{P(h = t)} \cdot P(h = t|\mathit{first}(rel) = a_1) \cdot P(h = t|sec(rel) = a_2)$$

If a_1 only rarely occurs in hyponym position in assumed hypernymy relations, we approximate p by $P(h = t|sec(rel) = a_2)$, analogously for rarely occurring concepts in the hypernym position. As usual, the probabilities are estimated by relative frequencies relying on a human annotation.

First Sentence: The first sentence of a Wikipedia article normally contains a concept definition and thus often expresses a hypernymy relation. Thus, the feature *First Sentence* is set to one, if the associated relation was extracted from a first sentence of a Wikipedia article at least once.

Frequency: The feature *frequency* regards the quotient of the occurrences of the hyponym in other extracted relation in hyponym position and the hypernym in hypernym position. The correlation of this feature with the confidence score is given in Table 1.

This feature is based on two assumption. First, we assume that general terms normally occur more frequently in large text corpora than very specific ones [15]. Second, we assume that usually a hypernym has more hyponyms than vice-versa [9, p.436–437]. Let us consider a simple example. The concept *city* occurs much more often in large text corpora than most cities in the worlds. Furthermore, the number of hyponyms of *city* is very large, since every city in the world is a hyponym of *city*, while the list of hypernyms of a certain city just contains a few concepts like *city*, *location* and *entity*. Therefore, the concept *city* is expected to occur much more often in a hypernym position of an extracted relation than a certain city in the hyponym position. Actually, most cities only occur at most once in an extracted hyponym relation from Wikipedia.

Context: Usually, the hyponym can appear in the same textual context as its hypernym[5]. The textual context can be described as a set of other concepts (or words for shallow approaches) which occur in the neighborhood of the regarded hyponym/hypernym. Analogously to Cimiano, we estimate the semantic similarity between hyponym and hyponym by:

$$hyponym(c_2, c_1) = \frac{|context(c_1) \cap context(c_2)|}{|context(c_1)|} \tag{7}$$

Instead of regarding textual context we investigate the possible properties which can occur at a $PROP$ edge leading from a concept in the SN. This has the advantages that a Word Sense Disambiguation (WSD) was already done and the association between the property and the concept was already established automatically by the SN which may not be trivial if the adjective which is associated to the property is used predicatively.

Pattern Features: For each pattern, an associated pattern feature is defined which is assigned the value one if the relation was extracted by this pattern, otherwise zero. Naturally, the same hypernymy relation can be determined by several patterns. The most strongly correlated pattern features were the feature related to the shallow pattern NP_{hypo} *is a* NP_{hyper} and the deep pattern D_1 shown in Equation 2. Note that in order to get an acceptable recall the pattern NP_{hypo} *is a* NP_{hyper} is only applied on the first sentences of Wikipedia articles.

6 EVALUATION

We applied the patterns on the German Wikipedia corpus from November 2006 which contains 500 000 articles. In total we extracted 391 153 different hypernymy relations employing 22 deep and 19 shallow patterns. The deep patterns were matched to the SN representation, the shallow patterns to the tokens. Concept pairs which were also recognized by the compound analysis were excluded from the results since such pairs can be recognized on the fly and need not be stored in the knowledge base. Thus, these concept pairs are disregarded for the evaluation. Otherwise, recall and precision would increase considerably.

We assigned each extracted concept pair a score calculated by the probability score for relation correctness estimated by a Support Vector Machine[16]. Furthermore, the correlation of all features to relation correctness (1.0 if relation is correct, 0.0 if incorrect) were determined, where a selection of that features is given in Table 1.

The correctness of an extracted relation is given for several confidence score intervals in Table 2 and Fig. 3. There are 89 944 concept pairs with a score of more than 0.7, 3 558 of them were annotated with the information of whether the hypernymy relation actually holds. Note that an extracted relation pair is only annotated as correct if it can be stored in a knowledge base without modification (except from redundancy removal). Thus, a relation is also considered incorrect if

- multi-token expressions are not correctly recognized,
- the singular forms of unknown concepts appearing in plural form are not estimated correctly (this is not trivial for the German language),
- the hypernym is too general, e.g., *word* or *concept*, or
- the wrong reading is chosen by the Word Sense Disambiguation.

We also investigated in which cases deep or shallow patterns were better applicable. Shallow patterns are applied on the tokenizer information of WOCADI. Naturally, shallow patterns are applicable even if a deep parse was not successful or the sentence was incorrectly parsed. In about 40% of all sentences, a complete SN could not be constructed which is caused either by unknown words, misspellings, grammatical errors or complex grammatical sentence structures.

In contrast, deep patterns are able to extract relations even if additional constituents are located between hyponym and hypernym which is often not possible using shallow pattern. For instance the shallow pattern *X bezeichnet ein Y 'X denotes an Y'* cannot be used to extract the relation $SUB(\mathit{bajonett.1.1}, \mathit{stoßwaffe.1.1})$ $(SUB(\mathit{bayonet.1.1}, \mathit{weapon.1.1}))$ from the sentence *Bajonett bezeichnet eine auf den Gewehrschaft aufsteckbare Stoßwaffe. 'literally: Bajonet denotes an on the gun stickable weapon.'* while this is possible for the deep pattern

$$\begin{aligned} D_3 : SUB(A,B) \leftarrow\ & SCAR(C,D) \wedge SUB(D,A) \wedge \\ & SUBS(C, \mathit{bezeichnen.1.2}(\mathit{denote})) \wedge \\ & OBJ(C,E) \wedge SUB(E,B) \end{aligned} \tag{8}$$

Similar considerations hold for the sentence: *Sein Geburtshaus in Marktl ist dasselbe Gebäude, in dem auch Papst Benedikt XVI. zur Welt kam. 'His house of birth in Marktl is the same building in which Pope Benedikt XVI. was born.'*

To handle all such cases with only shallow patterns would require the definition of a tremendous amount of patterns and is therefore not realistically possible in practice.

An example where the normalization from different surface representations and syntactical structures to a single SN proved to be useful:
Auf jeden Fall sind nicht alle Vorfälle aus dem Bermudadreieck oder aus anderen Weltgegenden vollständig geklärt. 'In any case, not all incidents from the Bermuda Triangle or from other world areas are fully explained.'
From the last sentence pair, a hypernymy pair can be extracted by application of rule D_1 (Equation 2) but not by any shallow patterns. The

Table 1. Correlation of features to relation correctness.

Feature	Correlation
Correctness Rate	0.207
Frequency	0.167
Context	0.084
Deep pattern D_1	0.077
Pattern NP_{hypo} *is a* NP_{hyper}	0.074

Table 2. Precision of the extracted hypernymy relations for different confidence score intervals.

Score	≥0.95	≥0.90	≥0.85	≥0.80	≥0.75	≥0.70	≥0.65	≥0.60	≥0.55
Correctness (%)	100.00	87.23	86.49	82.48	82.03	70.49	67.81	57.41	57.03

application of the shallow Hearst pattern NP_{hypo} $\{,NP_{hypo}\}*\{,\}$ *und andere/and other* NP_{hyper} fails due to the word *aus 'from'* which cannot be matched. To extract this relation by means of shallow patterns an additional pattern would have to be introduced. This could also be the case if syntactic patterns were used instead since the coordination of *Bermudadreieck 'Bermuda Triangle'* and *Weltgegenden 'word areas'* is not represented in the syntactic constituency tree but only on a semantic level [4].

149 900 of the extracted relations were only determined by the deep but not by the shallow patterns. If relations extracted by one rather unreliable pattern are disregarded, this number is reduced to 100 342. The other way around, 217 548 of the relations were determined by the shallow but not by the deep patterns. 23 705 of the relations were recognized by both deep and shallow patterns. Naturally, only a small fraction of the relations were checked for correctness. In total 6 932 relations originating from the application of shallow patterns were annotated, 4 727 were specified as correct. In contrast, 5 626 relations originating from the application of deep patterns were annotated and 2 705 were specified as correct.

7 Conclusion and Outlook

An approach was introduced for extracting hyponyms by a deep semantic approach. Instead of using the surface representation of sentences, the patterns are defined on a semantic level and are applied on SNs. The SNs are derived by a deep syntactico-semantic analysis. This approach is combined by a shallow method to guarantee an acceptable recall if sentences are not parsable. The evaluation showed that the recall could be considerably improved. In contrast to a shallow representation, the semantic patterns have the advantage of a greater generality which reduces the number of patterns. Furthermore, anthroponyms are already identified and parsed

[4] Note that some dependency parsers employ a semantic-oriented normalization too.

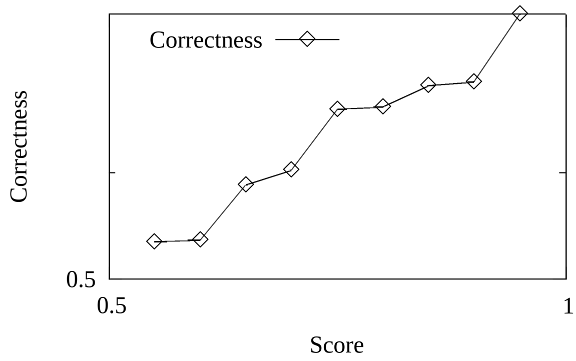

Fig. 3. Precision of the extracted hypernymy relations for different confidence score intervals.

by the SN which simplifies the extraction of instance-of-relations concerning person names.

Further possible improvements are the extraction of other semantic relations using this approach, for instance meronyms or antonyms. Furthermore, validation techniques will be further extended. We plan the usage of the ESPRESSO algorithm [17] as an additional feature and the employment of several deep features.

8 AKNOWLEDGEMENTS

We want to thank our colleagues of our department for their support. Especially we are indebted to I. Glöckner, C. Eichhorn, A. Pils and P. Grimm for proof-reading this work. This work is partly funded by the DFG project *Semantische Duplikatserkennung mithilfe von Textual Entailment* (HE 2847/11-1).

REFERENCES

1. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proc. of COLING, Nantes, France (1992)
2. Morin, E., Jaquemin, C.: Automatic acquisition and expansion of hypernym links. Computers and the Humanities **38**(4) (2004) 363–396
3. Snow, R., et al.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, Massachusetts (2005) 1297–1304

4. Sombatsrisomboon, R., et al.: Acquisition of hypernyms and hyponyms from the WWW. In: Proc. of the International Workshop on Active Mining, Maebashi City, Japan (2003)
5. Cimiano, P., et al.: Learning taxonomic relations from heterogeneous sources of evidence. In Buitelaar, P., et al., eds.: Ontology Learning from text: Methods, evaluation and applications. IOS Press, Amsterdam, Netherlands (2005) 59–73
6. Etzioni, O., et al.: Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence **165**(1) (2005) 91–134
7. Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. In: Proc. of ACL, Ann Arbor, Michigan (2005) 419–426
8. Quan, T., et al.: Automatic generation of ontology for scholarly semantic web. In: The Semantic Web - ISWC 2004. Volume 4061 of LNCS. Springer, Berlin, Germany (2004) 726–740
9. Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Springer, Berlin, Germany (2006)
10. Fellbaum, C., ed.: WordNet An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts (1998)
11. Hamp, B., Feldweg, H.: Germanet - a lexical-semantic net for german. In: Proc. of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, Madrid, Spain (1997)
12. Hartrumpf, S., Helbig, H., Osswald, R.: The semantically based computer lexicon HaGenLex – Structure and technological environment. Traitement automatique des langues **44**(2) (2003) 81–105
13. Hartrumpf, S.: Hybrid Disambiguation in Natural Language Analysis. PhD thesis, FernUniversität in Hagen, Fachbereich Informatik, Hagen, Germany (2002)
14. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
15. Joho, H., Sanderson, M.: Document frequency and term specificity. In: Proc. of RIAO, Pittsburgh, Pennsylvania (2007)
16. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. (2001)
17. Pantel, P., Pennachiottti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proc. of COLING/ACL, Sydney, Australia (2006)

TIM VOR DER BRÜCK
INTELLIGENT INFORMATION AND
COMMUNICATION SYSTEMS (IICS)
FERNUNIVERSITÄT IN HAGEN, 58084 HAGEN, GERMANY
E-MAIL: <TIM.VORDERBRUECK@FERNUNI-HAGEN.DE>