Hypernymy Extraction Based on Shallow and Deep Patterns

Tim vor der Brück

Intelligent Information and Communication Systems (IICS)
FernUniversität in Hagen
tim.vorderbrueck@fernuni-hagen.de
58084 Hagen, Germany

Abstract. There exist various approaches to construct taxonomies by text mining. Usually these approaches are based on supervised learning and extract in a first step 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 syntactic tree structure, i.e., a constituency or dependency tree derived by a syntactical parser. In this work we present an approach which, additionally to shallow patterns, directly operates on semantic networks which are derived by a deep linguistic syntactico-semantic analysis. Furthermore, the shallow approach heavily depends on semantic information, too. It is shown that recall and precision can be improved considerably than by relying on shallow patterns alone.

1 Introduction

A large knowledge base is needed by many tasks in the area of natural language processing, including question answering, textual entailment or information retrieval. One of the most important relations is *hypernymy* which is often referred to as the *is-a relation*. Quite a lot of effort was spent on hypernymy extraction from 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

A quite popular approach of the first type of algorithms was proposed by Hearst and consists of the usage of so–called Hearst patterns[1].

These patterns are applied on arbitrary texts and the instantiated pairs are then extracted as hypernymy relations. Several approaches were developed to extract such patterns automatically from a text corpus by either employing a surface [2, 3] or a syntactical tree representation [4].

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 a set of the words which frequently occur together with the hypernym (or hyponym). Whether a word is the hypernym of a second word can then be determined by a semantic similarity measure on the two sets [5]. If Word Sense Disambiguation is used, those approaches can operate directly on concepts instead of words which is currently rather rarely done.

A further method to extract hypernymy relations 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[6].

In this work[1] we will follow a hybrid approach. On the one side, we apply shallow patterns which do not require parsing but only need a tokenization of the analyzed sentence. In contrast to most common approaches our shallow method extracts pairs of concepts, not of words, as determined by Word Sense Disambiguation.

On the other side, we employ deep patterns directly on the semantic networks (SN) which are created by a deep semantic parser. These patterns are partly learned by text mining on the SN representations and partly manually defined.

We use for the extraction of hypernyms the German Wikipedia corpus from November 2006 which consists of about 500 000 articles.

2 System Architecture

Fig. 1 shows the architecture of our system *SemQuire* (*SemQuire* relating to *Acquire Semantic Knowledge*). In the first step, the Wikipedia corpus is parsed by the deep analyzer WOCADI [2] [7]. The parsing process does not employ a grammar but is based on a word class functional analysis. For that it uses a semantic lexicon [8] containing currently 28 000 deep and 75 000 shallow entries.

For each sentence, WOCADI tries to create a token list, a dependency tree and a SN. In contrast to the SN and the dependency tree, the token list is always created even if the analyzed sentence is ill-formed and not syntactically correct.

Both types of patterns (shallow and deep) are applied to the parse result of Wikipedia. In particular, the shallow patterns are applied on the token information while the deep patterns are applied on the SNs. If an application of such a pattern is successful, the variables occurring in the patterns are instantiated with concepts of the SN (or with concepts occurring in the token list for shallow patterns) and a hypernymy relation is extracted. Furthermore, a first validation is made which is based on semantic features and ontological sorts (a description of semantic features and ontological sorts can be found in [9]). If the validation is successful, the extracted relation is stored in the knowledge base. We currently develop an approach to generate for each relation a quality score by the combination of several features. Relations assigned a low quality score should then be

[1] Note that this work is related to the DFG-project: Semantische Duplikatserkennung mithilfe von Textual Entailment (HE 2847/11-1)

[2] WOCADI is the abbreviation for WOrd ClAss DIsambiguation.

seen with caution and have to be validated before using them in any operational system.

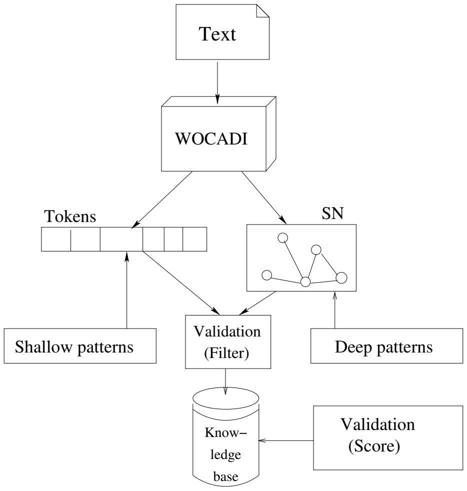

Fig. 1. System architecture of SemQuire.

3 Application of Shallow Patterns

The information for a single token as returned by the WOCADI parser consists of

- word-id: the number of the token
- char-id: the character position of the token in the surface string
- cat: the grammatical category
- lemma: a list of possible lemmas
- reading: a list of possible concepts
- parse-reading/lemma: a concept and lemma determined by Word Sense Disambiguation (see Fig. 2). The chosen concept must be contained in the concept list, analogously for the lemma. Note that concepts are marked by trailing numbers indicating the intended reading (e.g., *house.1.1*).

A pattern is given by a premise and a conclusion $SUB(a, b)$. The premise consists of a regular expression containing variables and feature value structures (see Fig. 2) where the variables are restricted to the two appearing in the conclusion (a,b). As usual, a question mark denotes the fact that the following expression is optional, a wildcard denotes the fact that zero or more of the following expression are allowed. The variables are instantiated with concepts relating to nouns from the token list (parse-lemma) as returned by WOCADI. The feature

```
(analysis-ml (
((word "Der") (word-id 1) (char-id 0) (cat (art dempro)) (lemma
("der")) (reading ("der.1" "der.4.1") (parse-lemma "der")
(parse-reading "der.1")))

((word "Bundeskanzler") (word-id 2) (char-id 4) (cat (n)) (lemma
("Bundeskanzler")) (reading ("bundeskanzler.1.1")) (parse-lemma
"bundeskanzler") (parse-reading "bundeskanzler.1.1"))

((word "und") (word-id 3) (char-id 18) (cat (conjc)) (lemma
("und")) (reading ("und.1")))

((word "andere") (word-id 4) (char-id 22) (cat (a indefpro))
(lemma ("ander")) (reading ("ander.1.1" "ander.2.1")) (parse-lemma
"ander") (parse-reading "ander.1.1"))

((word "Politiker") (word-id 5) (char-id 29) (cat (n)) (lemma
("Politiker")) (reading ("politiker.1.1")) (parse-lemma
"politiker") (parse-reading "politiker.1.1"))

((word "kritisierten") (word-id 6) (char-id 39) (cat (a v))
(lemma ("kritisieren" "kritisiert")) (reading ("kritisieren.1.1"))
(parse-lemma "kritisieren") (parse-reading "kritisieren.1.1"))

((word "das") (word-id 7) (char-id 52) (cat (art dempro)) (lemma
("der")) (reading ("der.1" "der.4.1")) (parse-lemma "der")
(parse-reading "der.1"))

((word "Gesetz") (word-id 8) (char-id 59) (cat (n)) (lemma
("Gesetz")) (reading ("gesetz.1.1")) (parse-lemma "Gesetz")
(parse-reading "gesetz.1.1"))

((word ".") (word-id 9) (char-id 67) (cat (period)) (lemma ("."))
(reading ("period.1")))))
```

Fig. 2. Token information for the sentence Der Bundeskanzler und andere Politiker kritisierten das Gesetz. *'The chancellor and other politicians criticized the law.'* as returned by the WOCADI parser.

```
(((SUB a b))
( a
( ((word ","))
  ? (((cat (art))))
  a)
((word "und"))
? (((cat (art))))
((word "andere"))
? (((cat (a))))
b))
```

Fig. 3. One shallow pattern used to extract hypernymy relations.

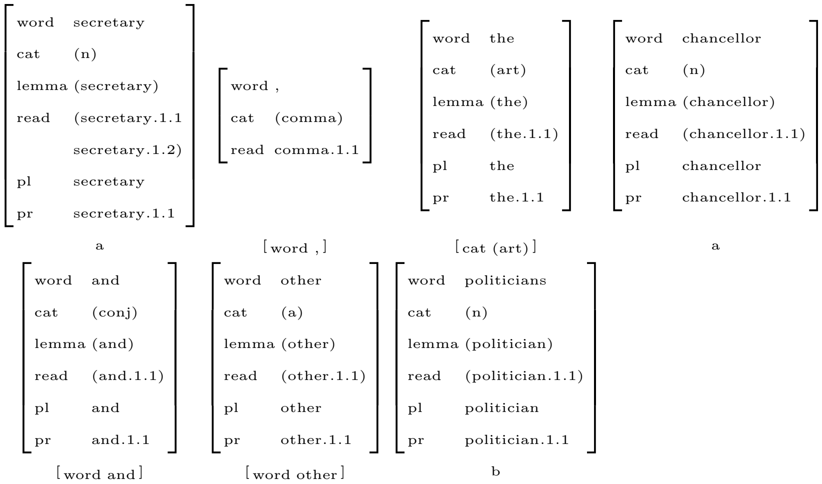

Fig. 4. Matching a pattern with a token list by unification. The pattern is displayed below the token information. Each variable is set to the value of the *parse-reading* attribute (*pr*=*parse-reading*, *read*=*reading*, *pl*=*parse-lemma*).

value structures are tried to be unified with token information from the token list. Since all variables of the conclusion must show up in the premise too, the premise variables are fully instantiated if a match is successful. The instantiated conclusion is then extracted as a hypernymy relation. Note that if a parse is not successful, a disambiguation to a single concept for a token is usually not possible. In this case the concept is chosen from the token's concept list which occurs in the corpus most often.

The entire procedure is illustrated in Fig. 4. If a variable appears several times in the premise part it is bound to several constants and, in the case a match could be established, the Cartesian product of all variable bindings for the two variables are extracted as relation pairs.
Example: *The chancellor, the secretary and other politicians criticized the law.* If the pattern specified in Fig. 4 is applied on the sentence above the vari-

able a can be bound to *chancellor.1.1* and *secretary.1.1*, b can be bound to *politician.1.1*. Thus, the two relations $SUB(\mathit{chancellor.1.1}, \mathit{politician.1.1})$ and $SUB(\mathit{secretary.1.1}, \mathit{politician.1.1})$ are extracted.

We employed 20 shallow patterns. A selection of them is displayed in Table 1[3]. Each pattern in this table is accompanied by a precision value which specifies the relative frequency that a relation extracted by this pattern is actually correct. Relations which are automatically filtered out by the validation component (see Sect. 2) are disregarded for determining the precision. The patterns s_3, s_5, s_7, s_8, s_9, and s_{10} are basically German translations of Hearst patterns. Note that pattern s_2, in order to get an acceptable precision, is only applied to the first sentences of Wikipedia articles since such sentences usually contain concepts related in a hypernymy relation.

4 Application of Deep Patterns

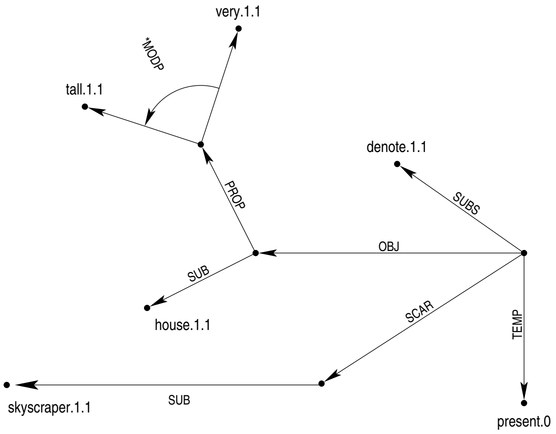

Fig. 5. SN for the sentence: *A skyscraper denotes a very tall house.*

In addition to shallow patterns we also employ several deep patterns. A selection of deep patterns is shown in Table 2.

Fig. 5 shows an example for an SN following the MultiNet paradigm[9]. An SN consists of nodes representing concepts and edges representing relations between concepts.

In addition, nodes can also be connected by means of functions (marked by a preceding *). In contrast to relations, the number of arguments is often

[3] Note that the patterns are actually defined as attribute value structures. However for better readability and space constraints we use a more compact representation in this table.

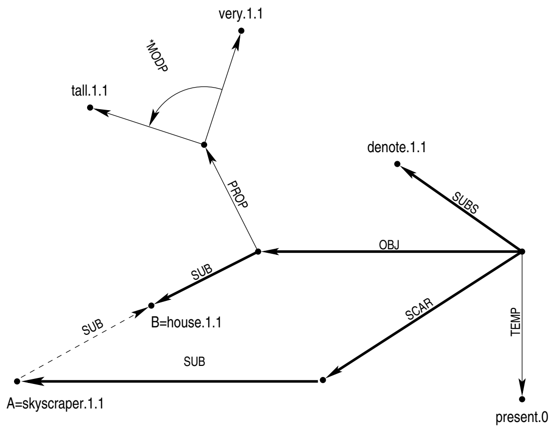

Fig. 6. SN matched with the pattern $SUB(A, B) \leftarrow SCAR(C, D) \wedge SUB(D, A) \wedge SUBS(C, denote.1.1) \wedge OBJ(C, E) \wedge SUB(E, B)$. Matching edges are printed in bold. The dashed arc is the inferred new edge.

Table 1. The shallow patterns (name, definition, precision) employed for hypernymy extraction where $A/A_i (1 \leq i \leq n + 1)$ is the hyponym of B. The symbol ^{l} denotes the fact that the lemma is referred to instead of the word form. The precision is not given for patterns which could not been matched often enough for reliable estimation.

Name	Definition	English Translation	Precision
s_1	als^l A (...) bezeichnet man B	A (...) is called B	0.79
s_2^*	A (...) ist ein B	A (...) is a B	0.75
s_3	A_1,...,A_n und ander^l B	A_1,..., A_n and other B	0.71
s_4	B wie A	B like A	0.70
s_5	A_1,...,A_n oder ander^l B	A_1,...,A_n or other B	0.66
s_6	B wie beispielsweise $A_1, \ldots, A_n$ und|oder A_{n+1}	B like for example $A_1, \ldots, A_n$ and|or A_{n+1}	0.63
s_7	B, insbesondere $A_1, \ldots, A_n$ und|oder A_{n+1}	B, especially $A_1, \ldots, A_n$ and|or A_{n+1}	0.57
s_8	B, einschließlich A_1,...,A_n und|oder A_{n+1}	B including A_1,...,A_n and|or A_{n+1}	0.28
s_9	solch ein B wie $A_1, \ldots, A_n$ und|oder A_{n+1}	such a B like $A_1, \ldots, A_n$ and|or A_{n+1}	-
s_{10}	solch^l B wie $A_1, \ldots, A_n$ und|oder A_{n+1}	such a B like $A_1, \ldots, A_n$ und|oder A_{n+1}	-
s_{11}	alle B außer $A_1, \ldots, A_n$ und A_{n+1}	all B except $A_1, \ldots, A_n$ and A_{n+1}	-
s_{12}	alle B bis auf $A_1, \ldots, A_n$ und A_{n+1}	all A except $A_1, \ldots, A_n$ and A_{n+1}	-

*: pattern is only matched to the first sentence of each Wikipedia article.

Table 2. A selection of deep patterns. $F_r(a_1, a_2)$: a_1 is the first argument of function r and precedes a_2 in the argument list; $G_r(a_1, a_2)$: a_1 precedes a_2 in the argument list of function r; $H_r(a_1, a_2)$: a_1 immediately precedes a_2 in the argument list of function r.

Name	Definition	Precision
d_1	$\mathit{SUB}(A,B) \leftarrow \mathit{SUB}(C,A) \wedge \mathit{PRED}(E,B) \wedge F_{*\mathit{ITMS}}(D,C) \wedge F_{*\mathit{ITMS}}(D,E) \wedge H_{*\mathit{ITMS}}(C,E) \wedge \mathit{PROP}(E, \mathit{ander.1.1}(\mathit{other.1.1}))$	0.74
d_2	$\mathit{SUB}(A,B) \leftarrow \mathit{SUB}(C,A) \wedge \mathit{PRED}(E,B) \wedge F_{*\mathit{ITMS}}(D,C) \wedge F_{*\mathit{ITMS}}(D,E) \wedge G_{*\mathit{ITMS}}(C,E) \wedge \mathit{PROP}(E, \mathit{ander.1.1}(\mathit{other.1.1})) \wedge \neg \mathit{ATTCH}(J,C)$	0.74
d_3	$\mathit{SUB}(A,B) \leftarrow \mathit{SUB}(C,A) \wedge \mathit{PRED}(E,B) \wedge F_{*\mathit{ITMS}}(D,C) \wedge F_{*\mathit{ITMS}}(D,E) \wedge G_{*\mathit{ITMS}}(C,E) \wedge \mathit{PROP}(E, \mathit{ander.1.1}(\mathit{other.1.1})) \wedge \neg \mathit{REFER}(E, \mathit{DET})$	0.73
d_4	$\mathit{SUB}(A,B) \leftarrow \mathit{SUB}(C,A) \wedge \mathit{PRED}(E,B) \wedge F_{*\mathit{ITMS}}(D,C) \wedge F_{*\mathit{ITMS}}(D,E) \wedge G_{*\mathit{ITMS}}(C,E) \wedge \mathit{PROP}(E, \mathit{ander.1.1}(\mathit{other.1.1}))$	0.73
d_5	$\mathit{SUB}(A,B) \leftarrow \mathit{PRED}(C,B) \wedge \mathit{SUB}(E,A) \wedge F_{*\mathit{ALTN1}}(D,C) \wedge F_{*\mathit{ALTN1}}(D,E) \wedge \mathit{PROP}(C, \mathit{ander.1.1}(\mathit{other.1.1}))$	0.71
d_6	$\mathit{SUB}(A,B) \leftarrow \mathit{SUB}(C,B) \wedge \mathit{SUB}(D,A) \wedge \mathit{SUB}(D,C)$	0.66
d_7	$\mathit{SUB}(A,B) \leftarrow \mathit{SCAR}(C,D) \wedge \mathit{SUB}(D,A) \wedge \mathit{OBJ}(C,E) \wedge \mathit{SUB}(E,B) \wedge \mathit{SUBS}(C, \mathit{bezeichnen.1.1}(\mathit{denote.1.1}))$	0.60
d_8	$\mathit{SUB}(A,B) \leftarrow \mathit{ARG2}(D,C) \wedge \mathit{SUB}(C,A) \wedge \mathit{MCONT}(E,D) \wedge \mathit{SUB}(F, \mathit{man.1.1}(\mathit{one.1.1})) \wedge \mathit{SUBS}(E, \mathit{bezeichnen.1.1}(\mathit{denote.1.1})) \wedge \mathit{ARG1}(D,G) \wedge \mathit{PRED}(G,B) \wedge \mathit{AGT}(E,F)$	0.51
d_9	$\mathit{SUB}(A,B) \leftarrow \mathit{ARG1}(D,E) \wedge \mathit{ARG2}(D,F) \wedge \mathit{SUBR}(D, \mathit{equ.0}) \wedge \mathit{SUB}(E,A) \wedge \mathit{SUB}(F,B)$	0.17

variable for functions. The result and the arguments of a function corresponds to MultiNet nodes. The following MultiNet relations and functions are used in the diagram shown in Fig. 5:

- *SUB*: relation of conceptual subordination for objects (hypernymy)
- *TEMP*: relation specifying the temporal embedding of a situation
- *PROP*: relation between object and property
- *SUBS*: relation of conceptual subordination for situations (troponymy)
- *SCAR*: cognitive role: carrier of a state, associated to a situation
- *OBJ*: cognitive role: neutral object, associated to a situation
- $*MODP$: function modifying properties

Additionally, each node in the SN is associated with a list of layer features, i.e., degree of generality (GENER), determination of reference (REFER), variability (VARIA), facticity (fact), intensional quantification (QUANT), pre-extensional cardinality (CARD) and entity type (ETYPE). The patterns can refer to the layer features too. Currently, only the layer feature REFER is used in our patterns (see pattern d_3 in Table 2). This layer feature specifies if a concept is determinate (for instance by usage of a definite article or a demonstrative determiner) or indeterminate.

The pattern

$$\begin{aligned} \mathit{SUB}(A,B) \leftarrow\ & \mathit{SCAR}(C,D) \wedge \mathit{SUB}(D,A) \wedge \mathit{SUBS}(C, \mathit{denote.1.1}) \wedge \\ & \mathit{OBJ}(C,E) \wedge \mathit{SUB}(E,B) \end{aligned}$$

can be matched to the SN displayed in Fig. 5 to extract the relation $\mathit{SUB}(\mathit{skyscraper.1.1}, \mathit{house.1.1})$ as illustrated in Fig. 6.

Note that different sentences can lead to the same SN. For instance, the semantically equivalent sentences *He owns a piano, a cello and other instruments.* and *He owns a piano, a cello as well as other instruments.* lead to the same SN. Thus, the pattern d_4 of Table 2 can be used to extract the relations $\mathit{SUB}(\mathit{piano.1.1}, \mathit{instrument.1.1})$ and $\mathit{SUB}(\mathit{cello.1.1}, \mathit{instrument.1.1})$ from both sentences. In general, the number of patterns can be considerably reduced by using an SN in comparison to the employment of a surface or a syntactic representation.

Pattern d_1, d_2, and d_3 in Table 2 are stricter versions of pattern d_4 which lead to a slight increase in precision for patterns d_1 and d_2. Practically no improvement was observed for pattern d_3. d_1 requires the hypernym node to follow immediately the hyponym node. This prevents the extraction of $\mathit{SUB}(\mathit{cookies.1.1}, \mathit{milk\_product.1.1})$ in the sentence: *We bought cookies, butter, and other milk products.* d_2 disallows other concept nodes to attach to the hyponym node which can be used to further specialize the hyponym candidate like in the sentence: *His father and other gangsters ....*

The concept node belonging to *his father* is subordinated to *gangster* but this is not the case for *father.* Thus, in contrast to pattern d_4, the pattern d_2 would not extract

$\mathit{SUB}(\mathit{father.1.1}, \mathit{gangster.1.1})$ from this sentence. d_3 finally requires that the hypernym should not be referentially determined.

5 Evaluation

We applied the patterns on the German Wikipedia corpus from 2005 which contains 500 000 articles. In total, we extracted 160 410 hypernymy relations employing 12 deep and 20 shallow patterns. The deep patterns were matched to the SN representation, the shallow patterns to the tokens. Concept pairs which are also recognized by the morphological compound analysis (a compound is normally a hyponym of its primary concept) were excluded from the results since such pairs can be recognized on the fly and need not to be stored in the knowledge base. Otherwise, the number of extracted concept pairs would be much larger than 160 410.

Naturally, shallow patterns have the advantage that they are applicable if the parse fails. On the other hand, deep pattern are still applicable, if there are additional constituents and subclauses between hypo- and hypernyms which usually cannot be covered by shallow patterns. The following sentences from the Wikipedia corpus are typical examples where the hypernymy relationship could only be extracted using deep patterns (hyponym and hypernym are underlined):

Das typisch nordhessische Haufendorf liegt am Emsbach im historischen Chattengau, wurde im Zuge der hessischen Gebiets- und Verwaltungsreform am 1. Februar 1971 Stadtteil von Gudensberg, und hatte 2005 1 400 Einwohner. 'Haufendorf, a typical north Hessian village, is located at the Emsbach in the historical Chattengau, became, during the Hessian area and administration reform, a district of Gudensberg and had 1 400 inhabitants in 2005.'

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_5 from Table 2 but not by any shallow patterns. The application of pattern s_5 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 would also be the case if deep syntactic patterns were used instead since the coordination of *Bermudadreieck 'Bermuda Triangle'* and *Weltgegenden 'word areas'* is not represented in the syntactic dependency tree but only on a semantic level.

We evaluated the portion of extracted relations which are regarded correct for every pattern. Obvious mismatches which are recognized automatically by checking ontological sort and semantic features of hyponym/hypernym for subsumption.

An extracted relation is only considered correct if it makes sense to store this relation without modifications in an ontology or name list. This means, extracted relations assumed to express hypernymy are considered incorrect if

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

The precision for each pattern is shown in Table 1 for the shallow patterns and in Table 2 for the deep patterns.

77 870 of the extracted relations were only determined by the deep but not by the shallow patterns. If relations extracted by the rather unreliable pattern d_9 are disregarded, this number reduces to 27 999. The other way around, 61 998 of the relations were determined by the shallow but not by the deep patterns. 20 542 of the relations were both recognized by deep and shallow patterns. Naturally, only a small fraction of the relations were manually checked for correctness. The accuracy of the annotated relations extracted by the shallow patterns is 0.62, by the deep ones 0.51. The accuracy of the relations extracted by both the deep and the shallow patterns is 0.80, considerably larger than the other two values.

6 Conclusion and Outlook

We introduced an approach for extracting hypernymy relation by a combination of shallow and deep patterns, where the shallow patterns are applied on the token list and the deep patterns on the SNs representing the meaning of the sentences. By using a semantic representation the number of patterns can be reduced in comparison to a syntactic or surface representation. Furthermore, by combining shallow and deep patterns the precision or the recall regarding the number of extracted relations can be improved considerably. If a parse was not successful, we still can extract relations by employing the shallow patterns. In contrast, if additional constituents show up between the hyponym and the hypernym, the application of shallow patterns often fails and the hypernymy relation can be extracted by the application of a deep pattern.

In order to further improve recall and precision, we currently work on assigning a quality score to the extracted hypernymy pairs.

Furthermore, the possibility that shallow patterns can require certain lemmas or concepts to show up in the token list is only rarely used and should be considered more often. In strong inflecting languages like German the usage of lemmas or concepts instead of word forms can improve the applicability of patterns considerably. Extracting relation using shallow patterns currently lead to a higher recall than for the deep ones. Thus, the collection of applied deep patterns should be further extended.

Finally, it is planned to transfer the entire relation extraction approach to other relations than hyponymy, especially meronymy. To do this, the shallow and deep patterns would have to be replaced. Several patterns for meronymy extraction are for instance described by Girju et al[10]. Furthermore, the two validation components (see Sect. 2) need to be modified for this purpose.

7 Acknowledgements

We wish to thank all people who contributed to this work, especially Sven Hartrumpf for proof-reading this document.

References

1. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics (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. Maria Ruiz-Casado, E.A., Castells, P.: Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In: 10th International Conference on Applications of Natural Language to Information Systems, Alicante, Spain (2005) 67–79
4. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, Massachusetts (2005) 1297–1304
5. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning taxonomic relations from heterogeneous sources of evidence. In Buitelaar, P., Cimiano, P., Magnini, B., eds.: Ontology Learning from Text: Methods, evaluation and applications. IOS Press, Amsterdam, The Netherlands (2005) 59–73
6. Quan, T.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic generation of ontology for scholarly semantic web. In: The Semantic Web - ISWC 2004. Volume 4061 of LNCS. Springer, Berlin, Germany (2004) 726–740
7. Hartrumpf, S.: Hybrid Disambiguation in Natural Language Analysis. PhD thesis, FernUniversität in Hagen, Fachbereich Informatik, Hagen, Germany (2002)
8. 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
9. Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Springer, Berlin, Germany (2006)
10. Girju, R., Badulescu, A., Moldovan, D.: Automatic discovery of part-whole relations. Computational Linguistics **32**(1) (2006) 83–135