Tim vor der Brück, Hermann Helbig

Meronymy Extraction Using An Automated Theorem Prover

In this paper we present a truly semantic-oriented approach for meronymy relation extraction. It directly operates, instead of syntactic trees or surface representations, on semantic networks (SNs). These SNs are derived from texts (in our case, the German Wikipedia) by a deep linguistic syntactico-semantic analysis. The extraction of meronym/holonym pairs is carried out by using, among other components, an automated theorem prover, whose work is based on a set of logical axioms. The corresponding algorithm is combined with a shallow approach enriched with semantic information. Through the employment of logical methods, the recall and precision of the semantic patterns pertinent to the extracted relations can be increased considerably.

1 Introduction

In most cases, objects are not elementary, rather composed of smaller objects, e.g., a car consists of wheels, windows, a gearshift, etc. Similarly, a group can be split up into its elements, e.g., a soccer team is composed of soccer players. Theses types of relationships are called meronymy. The whole or set is called the holonym while the corresponding part or element is called meronym.

Meronymy relations are required for a multitude of tasks in natural language processing, such as information retrieval or question answering. Let us consider a simple example. A user asks: “When was the last earthquake in Europe?”. If the knowledge base contains the dates of recent earthquakes for all countries and also the information which countries are part (meronyms) of Europe, then this question can be answered.

To create large meronymy databases manually is very tedious and requires a lot of work. Thus, automatic approaches are preferable. A lot of approaches retrieve such relations by text mining. The first step is to develop a set of patterns. In the second step, these patterns are then applied to new texts, where they are used to recognize meronym/holonym pairs. Normally, these approaches only use surface or syntactical tree representations, i.e., constituency or dependency trees derived by a syntactical parser. They do not employ any semantic formalism and are therefore unable to incorporate background knowledge. Furthermore, they mostly extract meronyms between words and not word-readings. From this, it follows that results cannot be (directly) used in concept-based ontologies.

In this paper, a semantic approach is described which directly operates on SNs following the MultiNet (Helbig, 2006) formalism[1] in order to extract meronymy relations.

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

This approach takes a knowledge base and a set of logical axioms into account. To this end, the entire content of the German Wikipedia with more than 20 million sentences is transformed into SNs using the WOCADI parser (Hartrumpf, 2002). The extraction patterns defined on a semantic level are mainly derived from the patterns given in (Girju et al., 2006). It is combined with a method based on shallow patterns enriched with semantic information if present.

In the next section, we review existing work on meronymy extraction. Section 3 describes the MultiNet formalism which is our representation for texts. An overview about the application of semantic patterns based on the MultiNet formalism is presented in Section 4. Section 5 describes how to incorporate a set of logical axioms and a knowledge base. The validation of extracted meronymy hypotheses is presented in Section 6. Section 7 illustrates how the correct meronymy subrelation can be selected. The architecture of our meronymy extraction system is given in Section 8. Evaluation results are specified in Section 9. Finally, the conclusion and an outlook of future work is given in Section 10.

2 Review of Existing Work

In this section, we give an overview on existing work on meronymy extraction. Quite popular are pattern-based approaches. Table 1 lists a collection of the patterns defined by Girju et al. (2006).

ID	Surface Pattern	Example
S_1	NP_{mero} is part of NP_{holo}	the **engine** is part of the **car**
S_2	NP_{holo}'s NP_{mero}	**girl**'s mouth
S_3	NP_{mero} of NP_{holo}	**eyes** of the **baby**
S_4	NP_{holo} verb:have NP_{mero}	The **table** has four **legs**.
S_5	NP_{holo} P NP_{mero}	A **bird** without **wings** cannot fly.
S_6	NP_{mero} P NP_{holo}	A **room** in the **house**.
S_7	$NP(N_{holo}\ N_{mero})$ (noun compound)	**door knob**
S_8	$NP(N_{mero}\ N_{holo})$ (noun compound)	**turkey pie**

Table 1: Some of the patterns suggested for the recognition of meronyms by Girju et al. (the lower indices mean: mero=meronym, holo=holonym)

These patterns are applied to arbitrary texts, and the instantiated variable NP_{mero} is extracted as meronym hypothesis of the assumed holonym NP_{holo}. Since hypotheses extracted by these patterns are not always correct, an additional validation component is required. Girju et al. employ a decision tree on annotated meronymy training data by making use of the WordNet hypernymy hierarchy.

Another pattern based approach is introduced by (Berland and Charniak, 1999). The validation of the extracted hypotheses is done by several statistical features taking the pattern by which a meronymy hypothesis was extracted into account as well as how likely the occurrence of the holonym hypothesis is if the given meronym hypothesis shows up. In contrast to the approach of Girju et al., this method is not supervised and needs no annotated data.

Some of the patterns are very often applicable but the extracted hypotheses are rarely correct. An example of such a pattern is NP_{holo}'s NP_{mero}, as proposed by Girju et al. (2006). The ESPRESSO system introduced a bootstrapping approach geared towards handling this problem in particular. The reliability scores of relation hypotheses are used to derive reliability scores for the patterns that extracted such hypotheses and vice-versa (Pantel and Pennacchiotti, 2006).

An alternative approach to pattern matching is the use of support vector machines and tree kernel functions, which are employed to assign one of several semantic relations including meronymy to a given word or concept pair. A tree kernel function is a function for comparing trees, where the matrix of kernel values is symmetric and positive-semidefinite. Such approaches follow the assumption that a certain semantic relation (e.g., meronymy) is quite likely to hold if there exist a lot of sentences with similar tree structures (or similar paths in the dependency trees) in which this relation is known to hold (Culotta and Sorenson, 2004; Bunescu and Mooney, 2005; Zhao and Grishman, 2005; Reichartz et al., 2009).

Current approaches for meronymy extraction as described above are practically neither semantic-based nor do they take background knowledge into account. Let us consider two examples which demonstrate how background knowledge can improve the evaluation results.

The following pattern is given:

$$\textsc{mero}(a1, a2) \leftarrow a1 \text{ is a member of } a2$$

This formula specifies that if a sentence contains the statement that *a1* is a member of *a2*, then *a1* is also a meronym (element) of *a2*. This pattern can be applied to the sentence *Mr. Peters is a member of AT&A.* to derive the meronymy relation[2] $\textsc{mero}(Mr.Peters, AT\&T)$. Now consider the sentence: *Mr. Peters is the leader of AT&T.* Naturally, the pattern is not applicable to this sentence. However, if we use background knowledge, then the fact that *Mr. Peters is a member of AT&T* can eventually be inferred by the fact that *Mr. Peters is the leader of AT&T*, which makes the pattern applicable. Thus, a large knowledge base can reduce the number of required patterns considerably and therefore the amount of work for the pattern developer. In this example the knowledge base was employed to improve the recall but it is also possible to improve precision.

Consider a sentence matching the surface pattern *x is a mixture of y and z* (e.g., *Water is a mixture of hydrogen and oxygen.*). Two meronymy relations can be extracted from such a sentence: $\textsc{mero}(y, x)$ and $\textsc{mero}(z, x)$ (in the example $\textsc{mero}(hydrogen, water)$ and $\textsc{mero}(oxygen, water)$). Note that the word mixture can also be used in a more abstract sense, e.g., *His attitude is a mixture of enthusiasm and diligence.* In order to prevent in this case the extraction of assumed meronymy rela-

[2] This expression is not a valid MultiNet expression but stated rather informally.

tions like MERO(*enthusiasm*, *attitude*) and MERO(*diligence*, *attitude*), one has to require that x and/or y/z are known to be hyponyms of *substance*, which can be expressed by a logical constraint taking the transitivity axiom of hyponymy into account. This example is described in more detail in Section 5.

Finally, a knowledge base can be advantageous for a multitude of other tasks, e.g., the majority of the axioms presented here are also used for question answering.

3 MultiNet as a Fine-grained Semantic Network Formalism

As described in Section 2, axioms can be used to make patterns more generally usable or to support the specifications of logical constraints. Naturally, in order to use logical axioms, all sentences have to be converted into a logical representation. We have decided to use the MultiNet SN formalism since this is a logical representation with great expressiveness (even beyond first order predicate logic) and is excellently supported by several software tools. In contrast to networks such as GermaNet (Hamp and Feldweg, 1997) or WordNet (Fellbaum, 1998), MultiNet is designed to represent both semantic and lexical relations between lexems as well as the meaning of whole natural language expressions. An SN consists of nodes representing concepts (word readings) and arcs denoting relations between concepts or functions involving concepts. In total there are approximately 120 relations and functions defined in MultiNet, including the following:

- ARG1/2: Specification of relational arguments at the metalevel
- ATTCH: Attachment of an objects to another object
- ATTR/VAL: Attribute-value specification
- ELMT: Element relation
- HSIT: Relation specifying the constituents of a hyper-situation
- *ITMS: Function enumerating a set
- MERO: Meronymy relation, hyper-relation of ELMT, HSIT, ORIGM^{-1}, PARS, SUBM, and TEMP
- ORIGM: Relation of material origin
- PARS: Meronymy relation except ELMT, HSIT, ORIGM^{-1}, SUBM, and TEMP
- *PMOD: Modification of objects by associative or operational properties
- PRED: Predicative concept characterizing a plurality
- SUB: Relation of subordination for conceptual objects (hyponym/instance of)
- SUB0: Relation of general hyponymy, hyper-relation of SUB, SUBR and SUBS
- SUBM: Set inclusion (subset)
- SUBR: Relation of conceptual subordination for relations
- SUBS: Relation of conceptual subordination for situations
- TEMP: Relation specifying the temporal embedding of a relation

In MultiNet, concepts are specified by a word label and a pair of indices $.n.m$ indicating the intended reading from a list of homographs or sememes of a polysemic word, respectively. These indices will henceforth be omitted from the text for the sake of brevity.

MultiNet is connected to the semantic lexicon HaGenLex (Hartrumpf et al., 2003). Each lexical entry of HaGenLex contains, aside from the typical morpho-syntactical information, one or more ontological sorts, a set of semantic features, and several layer features. The ontological sorts (currently more than 40) form a taxonomy. In contrast to other taxonomies, ontological sorts are not necessarily lexicalized, i.e., their names do not necessarily denote lexical entries.

The following list shows a small selection of ontological sorts:

- Object (o)
  - Concrete object (co): e.g., *milk*, *chair*
    - Discrete object (d): e.g., *chair*
    - Substance (s): e.g., *milk*, *honey*
  - Abstract object (ab): e.g., *thought*, *idea*
    - Abstract temporal object (ta): e.g., *month*
- Situation (si): e.g., *being warm*
- Quality (ql)
  - Property in the narrower sense (p): e.g., *tall*, *heavy*
  - Functional quality (fq): Such a quality obtains their full meaning only in connection with another entity.
    - Associative quality (aq), e.g., *chemical*, *philosophical*
    - Operational property (oq), e.g., *latter*, *third*

Semantic features denote semantic properties of objects; the values can be '+' (meaning applicable), '-' (not applicable) or 'underspecified'. A selection of semantic features is provided below:

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

Sample characteristics of the nominal concept *bear*:

- Ontological sort: d (discrete object)
- Semantic features: ANIMAL +, ANIMATE +, ARTIF -, HUMAN -, SPATIAL +, THCONC -, ...

In this paper we only employ the layer feature *type of extensionality* (*etype*). Therefore only this feature is described. It classifies nodes on the pre-extensional knowledge representation level (see (Helbig, 2006) or (Lyons, 2002) for a distinction of intensional and (pre)extensional interpretation) and can assume one of the following values:

0: Representative of the extensional of an elementary concept, which is itself not a set, e.g., *house*, *Max* (person named Max)
1: Set of elements of type 0, e.g., *several children*, *three cars*, *team*, *brigade*
2: Set of elements of type 1, e.g., *three crews*, *many organizations*, *umbrella organization*
3: Set of elements of type 2, e.g., *three umbrella organizations*

This list can theoretically be continued to arbitrary type numbers but only types of extensionality until the value of three are realistic in practice.

The networks expressed in the MultiNet formalism are obtained from surface texts by means of the syntactico-semantic parser WOCADI (Hartrumpf, 2002), based on a word-class controlled functional analysis. Note that, although a meronymy relation can be represented in the MultiNet formalism, it is usually not contained in the SNs which are created by the parser unless such a relation is already comprised in the knowledge base.

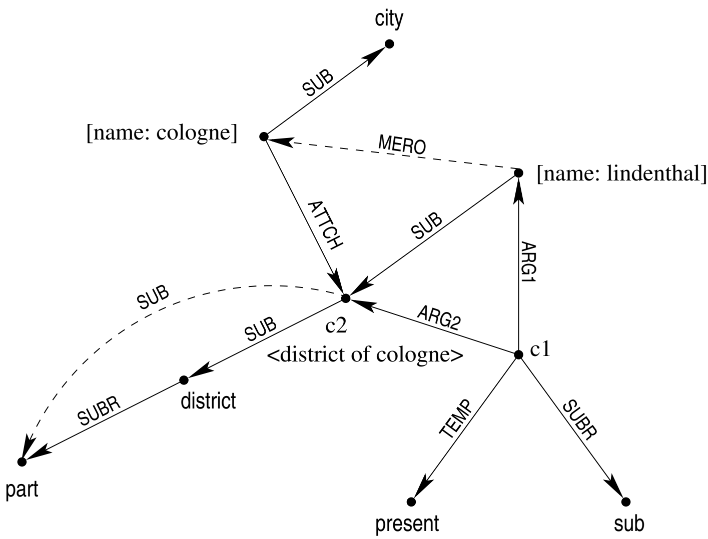

Figure 1: Application of the deep pattern D_4 to an SN representing the sentence: *Lindenthal is a district of Cologne.* Inferred edges are marked by dashed lines. $\textsc{subr}(c1, sub)$ indicates that the two arguments of $c1$ (c2 and c3) are connected by a SUB relation.

An example of an SN based on the MultiNet formalism is given[3] in Figure 1 and represents the sentence: *Lindenthal is a district of Cologne.* The specification of names (here Lindenthal and Cologne) is done by attribute/value constructs (MultiNet relations: ATTR and VAL). For better readability, an attribute value construct connected with a node c and subordinated to an attribute named n with value v, which is represented by the MultiNet expression $\textsc{attr}(c, d) \land \textsc{sub}(d, n) \land \textsc{val}(d, v)$, is written as $[n : v]$ in this figure. The parser recognized that a hyponymy relation is specified (SUBR(c1,sub)) and that the concept named *Lindenthal* is a hyponym (MultiNet relation: SUB) of the concept capsule associated to *district of Cologne*. The preposition *of* is realized by the relation ATTCH (attach) in the SN.

[3] Concept names are translated from German into English for better readability.

ID	Deep Pattern	Example
D_1	$\textsc{mero}(a1, a2) \leftarrow \textsc{subs}(d, consist) \wedge \textsc{arg1}(d, e) \wedge$ $\textsc{sub}(e, a2) \wedge \textsc{arg2}(d, f) \wedge$ $PAR_{*\textsc{itms}}(f, g) \wedge \textsc{pred}(g, a1)$	A car (a2) consists of wheels (a1),...
D_2	$\textsc{mero}(a1, a2) \leftarrow \textsc{sub}(c, a1) \wedge$ $\textsc{attch}(d, c) \wedge \textsc{sub}(d, a2)$	wheel(a1) of a car(a2)
D_3	$\textsc{mero}(a1, a2) \leftarrow \textsc{arg1}(e, d) \wedge \textsc{sub}(d, a2) \wedge$ $\textsc{arg2}(e, f) \wedge \textsc{sub}(f, mixture) \wedge$ $\textsc{attch}(g, f) \wedge PAR_{*\textsc{itms}}(g, h) \wedge \textsc{sub}(h, a1) \wedge$ $\textsc{subr}(e, sub) \wedge (\textsc{sub}(a2, substance) \vee$ $\textsc{sub}(a1, substance)) \wedge a2 \neq mixture$	Water (a2) is a mixture of hydrogen (a1) and ....
D_4	$N_1 := \textsc{attr}(a1, e) \wedge \textsc{sub}(e, name) \wedge \textsc{val}(e, d)$ $N_2 := \textsc{attr}(a2, g) \wedge \textsc{sub}(g, name) \wedge \textsc{val}(g, h)$ $\textsc{mero}(a1, a2) \wedge N_1 \wedge N_2 \leftarrow$ $N_1 \wedge N_2 \wedge$ $\textsc{arg1}(c, a1) \wedge \textsc{arg2}(c, f) \wedge$ $\textsc{sub}(f, part) \wedge \textsc{attch}(a2, f) \wedge$ $\textsc{subr}(c, sub)$	Germany (a1) is part of Europe (a2).
D_5	$\textsc{mero}(a1, a2) \leftarrow \textsc{arg1}(d, e) \wedge$ $PAR_{*\textsc{itms}}(e, f) \wedge \textsc{sub}(f, a1) \wedge$ $\textsc{arg2}(d, g) \wedge \textsc{pred}(g, part) \wedge$ $\textsc{attch}(h, g) \wedge \textsc{sub}(g, a2) \wedge$ $\textsc{subr}(d, equ)$	Wheels, windows and a roof (a1) are part of car (a2).
D_6	$\textsc{mero}(a1, a2) \leftarrow \textsc{sub}(d, member) \wedge$ $\textsc{attch}(e, d) \wedge \textsc{sub}(e, a2) \wedge \textsc{arg2}(f, d) \wedge$ $\textsc{arg1}(f, g) \wedge \textsc{sub}(g, d) \wedge \textsc{sub}(g, a1)$	A goalkeeper (a1) is a member of a soccer-team (a2).

Table 2: A selection of deep meronymy patterns formulated by means of MultiNet relations ($PAR_f(x, y)$ denotes the fact that x is the result of function f where one of the arguments of f is y).

4 Application of Deep Meronymy Patterns

The meronymy extraction process is based on semantic patterns (see Table 2). Each pattern consists of a premise and a conclusion $\textsc{mero}(a1, a2)$ (MERO is the MultiNet relation indicating meronymy) for generic concepts and $\textsc{mero}(a1, a2) \wedge N_1 \wedge N_2$ for instances where N_1 and N_2 are attribute/value constructs for $a1$ and $a2$. The premise is given as an SN. Two of the nodes in this SN should be labeled with $a1$ and $a2$ in order for the pattern be applicable.

Shallow pattern	Matching expression
$\textsc{mero}(a1,a2) \leftarrow$ a1 ((word "of")) ? (((cat (art)))) a2	wheel (a1) of a car (a2)
$\textsc{mero}(a1,a2) \leftarrow$ a2 ((word "with")) a1	house (a2) with balcony (a1)
$\textsc{mero}(a1,a2) \leftarrow$ a2 ((word "without")) a1	bird (a2) without wings (a1)
$\textsc{mero}(a1,a2) \leftarrow$ a1 ((word "is")) ((word "the")) ((word "main component") ((word "of")) a2	A CPU (a1) is the main component of computers (a2).
$\textsc{mero}(a1,a2) \leftarrow$ a1 * (((word ",")) ? (((cat (art)))) a1) ((word "and")) a1 ((word "are")) ((word "parts")) ((word "of")) ? (((cat (art)))) a2	CPU (a1), ...are parts of a computer (a2)
$\textsc{mero}(a1,a2) \leftarrow$ a1 * (((word ",")) ? (((cat (art)))) a1) ((word "and")) ? (((cat (art)))) a ((word "are")) ((word "components")) ((word "of")) ? (((cat (art)))) a2)	CPU (a1) and display card (a1) are components of computers (a2)
$\textsc{mero}(a1,a2) \leftarrow a2$ ((word "consists")) ((word "of")) a1 ? (* (((word ",")) a1) ((word "and")) a1))	a computer (a2) consists of transistors (a1),...

Table 3: Selection of shallow patterns for meronymy extraction. The expressions occurring in the patterns are translated from German to English.

The matching of the pattern with an SN is done by an automated theorem prover for first order predicate calculus. In comparison to ordinary pattern matching, this has the advantage that logical axioms can be included into the pattern-matching process. Note that this paper only describes the use of a theorem prover for extracting and not for validating meronymy hypotheses, which is, for instance done by Suchanek et al. (2009); vor der Brück and Stenzhorn (2010). In total, there are 19 deep patterns which are mainly derived from the patterns introduced by (Girju et al., 2006).

To apply a pattern of the form[4] $\textsc{mero}(a1,a2) \leftarrow \mathit{premise}$, where both $a1$ and $a2$ must show up in the premise, one has to find the bindings I for $a1$ and $a2$, which are

[4] Patterns for instances can be applied similarly.

required to cause the following formula to become a tautology:

$$\begin{aligned}&\textsc{mero}(a1^I, a2^I) \leftarrow \\ &((\textsc{mero}(a1^I, a2^I) \leftarrow premise^I) \wedge SNX)\end{aligned} \tag{1}$$

in which $SNX = SN \wedge KB$ (KB=knowledge base). The knowledge base contains a set of axioms as well as a large number of lexical and semantic relations. If variable bindings are determined successfully, than the relation $\textsc{mero}(a1^I, a2^I)$ is extracted as meronymy hypothesis.

The proof is found by deriving a contradiction (i.e., the empty clause) from the negated expression (1), using resolution:

$$\begin{aligned}\bot \equiv &(((\textsc{mero}(a1^I, a2^I) \leftarrow premise^I) \wedge SNX) \wedge \\ &\neg\textsc{mero}(a1^I, a2^I) \Leftrightarrow \\ &(\text{Use distributive law and } A \leftarrow B \equiv \neg B \vee A) \\ \bot \equiv &\neg premise^I \wedge SNX \wedge \neg\textsc{mero}(a1^I, a2^I) \Leftarrow \\ \bot \equiv &\neg premise^I \wedge SNX\end{aligned} \tag{2}$$

Thus, since the trivial case that $\textsc{mero}(a1^I, a2^I)$ already appears in the knowledge base, premise or SN should be disregarded, it is required to derive a contradiction from $\neg premise^I \wedge SNX$. This is done by showing that the empty clause can be obtained applying logical resolution on $\neg premise^I \wedge SNX$. For this purpose, the MultiNet theorem prover is employed, which is also successfully used in question-answering tasks (Glöckner, 2007) and is optimized for this SN scenario. In our tests, the MultiNet theorem prover was ten times faster than the well-known general purpose theorem prover E-KRHyper (Baumgartner et al., 2007).

For easier processing, functions with a variable number of arguments are converted into a set of binary relations. For each such function $x_p = f(x_1, \ldots, x_l)$ we create l relations $PAR_f(x_p, x_1)$,...,$PAR_f(x_p, x_l)$ to represent the parent-child relationships between the result and the arguments and $(l(l-1))/2$ relations to represent the sequence of the arguments: $FOLL_f(x_i, x_j) \Leftrightarrow i < j$. Thus, an example expression $res = {*}\textsc{itms}(x_1, x_2, x_3)$ (the MultiNet relation $*$ITMS combines several concepts in a conjunction) can be replaced by the following formula:

$$\begin{aligned}&\underset{*\textsc{itms}}{PAR}(res, x_1) \wedge \underset{*\textsc{itms}}{PAR}(res, x_2) \wedge \underset{*\textsc{itms}}{PAR}(res, x_3) \wedge \\ &\underset{*\textsc{itms}}{FOLL}(x_1, x_2) \wedge \underset{*\textsc{itms}}{FOLL}(x_1, x_3) \wedge \underset{*\textsc{itms}}{FOLL}(x_2, x_3)\end{aligned} \tag{3}$$

In addition to deep patterns, several shallow patterns are employed. Instead of a SN the premise consists of a regular expression involving feature value structures. The features are

- word: surface word form
- lemmas: possible lemmas

- categories: possible categories
- parse-lemma: lemma disambiguated by the Word Sense Disambiguation of the parser
- parse-reading: concept disambiguated by the Word Sense Disambiguation of the parser

These feature value structures are tried to be unified with the token information list provided by the parser. The most important employed shallow patterns are given in table 3. The applicability of these patterns could be further improved by adding additional optional tokens (like articles or adjectives). However, this make the patterns more difficult to read and also increases the extraction time. This is a drawback to deep patterns where such optional parameters are not needed.

5 Support via Logical Axioms

The use of an automated theorem prover together with the axiomatic apparatus of the MultiNet formalism has the advantage that the number of deep patterns can be considerably reduced compared with the number of shallow patterns. The axioms were mostly already developed for the task of question-answering and are reused for meronymy extraction.

ID	Axiom	# Hypotheses
A_1	$\textsc{sub}(x,s) \leftarrow \textsc{sub}(x,p) \wedge$ $*\textsc{pmod}(p,q,s) \wedge sort(q) = oq$	25 576
A_2	$\textsc{sub}(x,z) \leftarrow \textsc{sub0}(x,y) \wedge \textsc{sub}(y,z)$	14 258
A_3	$PAR_{*\textsc{itms}}(a,d) \wedge \textsc{pred}(d,c) \leftarrow$ $\textsc{pred}(a,c) \wedge \neg PAR_{*\textsc{itms}}(e,a)$	2 117
A_4	$\textsc{subs}(x,z) \leftarrow \textsc{subs}(x,y) \wedge$ $\textsc{subs}(y,z)$	564
A_5	$\textsc{attch}(a,e) \leftarrow \textsc{loc}(e,l) \wedge$ $\{*\textsc{in}(l,a) \vee *\textsc{at}(l,a)\} \wedge$ $\{\textsc{subs}(e,s) \vee \textsc{preds}(e,s)\}$	194
A_6	$\textsc{pred}(x,s) \leftarrow \textsc{pred}(x,p) \wedge$ $*\textsc{pmod}(p,q,s) \wedge sort(q) = oq$	81
A_7	$\textsc{sub}(a,s) \leftarrow \{\textsc{agt}(e,a) \vee \textsc{exp}(e,a) \vee$ $\textsc{mexp}(e,a)\} \wedge \textsc{ctxt}(e,c) \wedge \textsc{sub}(c,s)$	57

Table 4: Selected MultiNet axioms and the number of extracted hypotheses.

Table 4 presents the most successful axioms together with the number of hypotheses extractions for which a certain axiom was required. A_1 from Table 4 is required most often, where the function *PMOD is used to combine a conceptual object s with an operational property[5] (see Section 3), denoted by q in axiom A_1, which yields a more

[5] Operational properties having sort oq and associative properties are in contrast to properties in the narrower sense, which are treated by the MultiNet relation PROP (Helbig, 2006)).

special object p. This axiom is needed, for example, to deduce that *the first violinist* is a *violinist*.

Another example illustrating the usefulness of axioms is axiom A_2 (see Table 4). This axiom states the (generalized) transitivity of the hyponymy relation. Let us consider the SN representing the sentence *Lindenthal is a district of Cologne.* as displayed in Figure 1. Let us further assume the knowledge base contains the fact: $\textsc{subr}(\mathit{district}, \mathit{part})$. The concept represented by node *c2* is a hyponym of *district*. It can thus be inferred that *c2* is also a hyponym of *part*. Thus, the deep pattern D_4 (see Table 2) corresponding roughly to the surface pattern *a1 is a part of a2*, can also be applied to this SN. This means that the pattern D_4 becomes more generally usable by using axiom A_2.

A further frequently used axiom is

$$A_3 : \underset{*\textsc{itms}}{PAR}(a, d) \wedge \textsc{pred}(d, c) \leftarrow \textsc{pred}(a, c) \wedge \neg \underset{*\textsc{itms}}{PAR}(e, a) \tag{4}$$

This axiom is employed to add an $*$ITMS-relation to a PRED-edge, i.e., this axiom creates a coordination existing of exactly one element. In this way, a meronym inside and outside a coordination can be recognized by the same pattern.
Examples: **Car wheels** are part of a **car**. **Car wheels**, a **blinker** and a **gearshift** are part of a **car**. By applying axiom A_3 and pattern D_4, the meronymy relation $\textsc{mero}(\mathit{car\_wheel}, \mathit{car})$ can be extracted from both example sentences. The literal $\neg PAR_{*\textsc{itms}}(e, a)$ prevents the generated relation $\textsc{pred}(d, c)$ to be expanded by a further application of this axiom.

Axiom A_4 defines the transitivity of the SUBS-relation, and axiom A_5 states that an object located 'within' or 'at' another object is attached to this concept. A_6 is a variant of A_1; it only applies if pluralities (indicated by the MultiNet relation PRED) are considered. Finally A_7 is required if people act in certain roles (Example: *If someone does something as a father, then he is a father*).

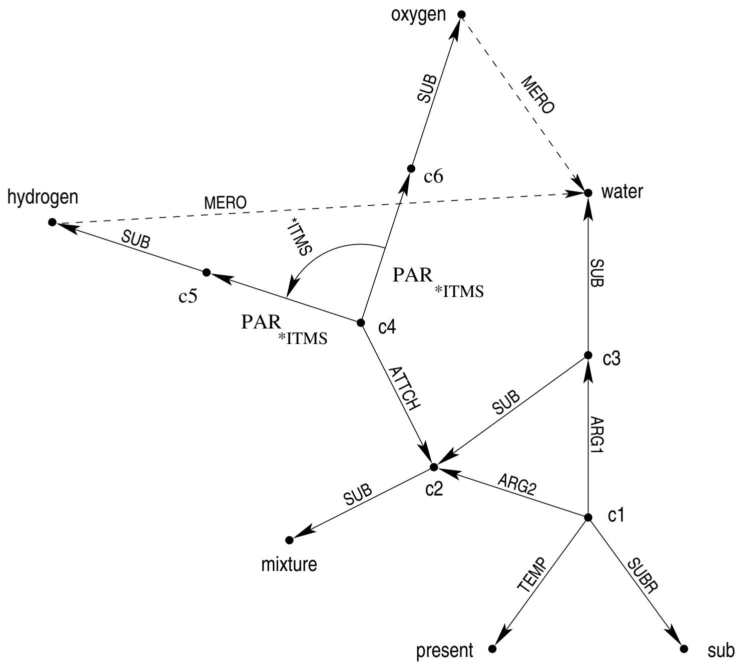

Figure 2: Application of the deep pattern D_3 to an SN representing the sentence: *Water is a mixture of hydrogen and oxygen.* The inferred edges are marked by dashed lines.

In the examples above, the use of axioms increases the recall. However, axioms can also help to increase the precision.

Consider the sentence *Water is a mixture of hydrogen and oxygen.* where the associated SN is given [6] in Figure 2. $\textsc{subr}(c1, sub)$ indicates that the two arguments of $c1$ (target concepts of $\textsc{arg}1$ and $\textsc{arg}2$) are connected by a SUB relation. The *ITMS function is used to combine the two components *hydrogen* and *oxygen* in a conjunction.

The two meronymy relations $\textsc{mero}(oxygen, water)$ and $\textsc{mero}(hydrogen, water)$ can be extracted from this sentence by applying pattern D_3 from Table 2 (roughly corresponding to the surface pattern given in Equation 5) to the associated SN.

$$\textsc{mero}(x, z) \wedge \textsc{mero}(y, z) \leftarrow z \text{ is a mixture of } x \text{ and } y \tag{5}$$

Note that the word mixture can also be used in a more abstract sense, e.g., *His attitude is a mixture of enthusiasm and diligence.* In order to prevent, in this

[6] Please note that c5 and c6 are generic nodes, and the arcs labeled with MERO should begin at c5 and c6, respectively. The latter is achieved in a post-processing step.

case, the extraction of assumed meronymy relations, like $\textsc{mero}(\textit{enthusiasm}, \textit{attitude})$ and $\textsc{mero}(\textit{diligence}, \textit{attitude})$, one has to require that at least one of y/z and x is known to be a *substance*, which is expressed by the disjunction $\textsc{sub}(\textit{a2}, \textit{substance}) \vee \textsc{sub}(\textit{a1}, \textit{substance})$ in pattern D_3. A disjunction is used instead of the conjunction $\textsc{sub}(\textit{a2}, \textit{substance}) \wedge \textsc{sub}(\textit{a1}, \textit{substance})$ since the lexical resources are limited and the hypernymy relations in the knowledge base are by no means complete. The pattern D_3 is applicable to the sentence *Water is a mixture of hydrogen and oxygen.* since water is a *substance*. The fact that water is a substance is derived by several applications of the axiom A_2 – *Transitivity of* SUB. The logical restriction $a_2 \neq \textit{mixture}$ is required in order to prevent the extraction of $\textsc{mero}(\textit{oxygen}, \textit{mixture})$ and $\textsc{mero}(\textit{hydrogen}, \textit{mixture})$.

This pattern is not applicable in the aforementioned example, where *mixture* is used in an abstract sense, because neither *attitude*, *diligence* nor *enthusiasm* are hyponyms of *substance*.

6 Validation

Not all of the extracted meronymy hypotheses extracted by deep or shallow patterns are correct. Thus, a validation component is required which checks each hypothesis for correctness by several semantic and statistical features.

The knowledge validation carried out is done in two steps. In the first step we compare the ontological sorts of relational arguments and semantic features to each other and filter out hypotheses of non-allowed combinations. In the second step the remaining hypotheses are assigned a confidence score which estimates their probability of correctness. This two step mechanism was chosen for performance reasons. In this way the size of the database containing the meronymy hypotheses and the runtime of the confidence score computation can be greatly reduced.

Since the confidence score represents a probability, hypotheses with a score of at least 0.5 are considered to be correct. All other hypotheses are considered incorrect. This score can be useful for two reasons:

- Hypotheses with a score beyond a certain score threshold could automatically be added to the knowledge base.
- If the hypotheses are to be validated manually, the annotator can first check the hypotheses with the high scores. In this way, he can add much more meronyms to the knowledge base in a given time interval than if he chooses the hypotheses randomly.

6.1 Filtering

Only certain combinations of ontological sorts and semantic features are permitted. For instance, a concept denoting a human being (semantic feature: human +) can only be meronym to another concept that denotes a set (recognizable in MultiNet by the type of extensionality). The type of extensionality is zero for an individual concept (which is not a set) and i for a set of elements of type $i-1$ for $i > 0$ (see Section 3). The ontological sorts of the meronym and holonym hypothesis must usually be identical (exception: sort s (substance) for meronym and d (discrete) for holonym or vice-versa are allowed).

The permitted combinations of ontological sorts are specified manually, while the regularities concerning features, which are a lot more complicated, are automatically learned by a tree augmented naïve Bayes algorithm (TAN) (Friedmann et al., 1997). The training is done separately for different combinations of the type of extensionality. The training data consists of a set of annotated meronymy relation hypotheses (meronym/no meronym) and lists of semantic features where the features are sought and found in the lexicon. Only relational candidates for whom semantic features and ontological sorts can be shown to be compatible are stored in the knowledge base.

6.2 Scoring

All hypotheses in the knowledge base are assigned a confidence score. This is done by means of a support vector machine (SVM) applied on several feature values[7] and annotations. The SVM (here LIBSVM, Chang and Lin (2001)) determines the class (meronymy or non-meronymy) and a probability estimate for each candidate pair and is trained on a set of annotated meronymy hypotheses. The annotation is either one (1) for meronymy or zero for non-meronymy. If the classification is 'meronymy', the score is defined by this probability estimate, otherwise as one minus this value. The employed features are explained below:

Use of a Taxonomy: This feature exploits a collection of known meronyms, hyponyms, synonyms, as well as a list of known non-meronym pairs. For that we used the lexical and semantic relations contained in HaGenLex. These relations are in part handcrafted, and in part derived from Wiktionary. Additionally, GermaNet relations, where the synsets are mapped to HaGenLex concept ids, are employed. This method works as follows: Consider a given pair of concepts $(a1, a2)$ being meronymically related to each other. Determine the Cartesian product $S(a1) \times S(a2)$ of all hypernyms of both normalized components (including the components itself). Increase a counter *pos* for all possible pairs of concepts in the resulting Cartesian product $S(a1) \times S(a2)$, i.e., set

$$\begin{aligned} pos(x,y) :=& pos(x,y) + 1 \\ & \forall x \in S(a1), y \in S(a2) \end{aligned} \tag{6}$$

where

$$S(x) = \{syno\_normalize(x)\} \cup \{syno\_normalize(z) | \textsc{sub}(x,z)\} \tag{7}$$

[7]not to be confused with semantic features

where *syno_normalize* : *Concepts* → *Concepts* is a function which maps a concept to the smallest element (according to some total ordering) of its synset:

$$syno\_normalize(c) = d :\Leftrightarrow \mathrm{SYNO}(d, c) \wedge (\forall e : \mathrm{SYNO}(e, d) \Rightarrow e \geq d) \tag{8}$$

Example: Assume *synset*(car)={auto,car}. Then *syno_normalize*(*car*) = *auto* if lexicographic ordering is used. By employing a synonymy normalization, the set of regarded concepts can be reduced which leads to a smaller memory consumption.

Analogously to *pos*, determine $neg(x, y)$ for all non-meronym pairs. For a new pair (x', y') opt for meronymy, iff

$$\begin{aligned} &\max_{a \in S(x'), b \in S(y')} \left\{ \frac{pos(a,b)}{pos(a,b)+neg(a,b)} \right\} > \\ &\max_{c \in S(x'), d \in S(y')} \left\{ \frac{neg(c,d)}{pos(c,d)+neg(c,d)} \right\} \end{aligned} \tag{9}$$

That means that the decision is in favor of meronymy iff there is a pair in $S(x') \times S(y')$ for which the indication for meronymy is stronger than the indication against meronymy for any other pair in $S(x') \times S(y')$. This decision procedure basically follows the approach presented by Costello (2007).

In addition to the approach of Costello, we employed the taxonomy to cancel out all meronymy hypotheses that appear in or can be derived from this taxonomy since a meronym cannot be a hyponym or a hypernym simultaneously. If one or both of the considered concepts are associated to compound words, then we also check for known hyponymy relations between all combinations of meronymy concept+base concept and holonymy+base concept (base concept: concept corresponding to the correct reading of the base word) candidates.

Correctness Rate: The feature *Correctness Rate* takes into account that the recognized holonym alone is already a strong indication for the correctness or incorrectness of the investigated hypothesis. The same holds for the assumed meronym as well. For instance, meronymy candidate pairs with the assumed holonym *computer*, *university*, or *building* were mostly correct. In contrast, meronymy candidate pairs with an assumed holonym *future* or *reason* were usually incorrect.

Thus, this indicator determines how often a concept pair is actually correct if a certain concept shows up in the first (meronym) or second (holonym) position. More formally, we are interested in determining the following probability:

$$p = P(me = t | arg1_r = a1 \wedge arg2_r = a2) \tag{10}$$

where

- $arg1_r$ denotes the first concept (the assumed meronym) in a given relation r.
- $arg2_r$ denotes the second concept (the assumed holonym) in a given relation r.
- $me(ronym) = t(rue)$ denotes the fact that a meronymy relation holds.

Applying Bayes' theorem to Equation 10 leads to the Equation:

$$p = P(me = t) \cdot \frac{P(arg1_r = a1 \wedge arg2_r = a2 | me = t)}{P(arg1_r = a1 \wedge arg2_r = a2)} \tag{11}$$

For better generalization, we assume that the events $arg1_r = a1$ and $arg2_r = a2$ as well as $(arg1_r = a1 | me = t)$ and $(arg2_r = a2 | me = t)$ are independent. Using these assumptions, Equation 11 can be rewritten:

$$\begin{aligned} p \approx p' &= P(me = t) \cdot \frac{P(arg1_r = a1 | me = t)}{P(arg1_r = a1)} \cdot \frac{P(arg2_r = a2 | me = t)}{P(arg2_r = a2)} \\ &= \frac{P(arg1_r = a1 \wedge me = t)}{P(arg1_r = a1)} \cdot \frac{P(arg2_r = a2 \wedge me = t)}{P(me = t) \cdot P(arg2_r = a2)} \\ p' &= \frac{1}{P(me = t)} \cdot P(me = t | arg1_r = a1) \cdot P(me = t | arg2_r = a2) \end{aligned} \tag{12}$$

If $a1$ only rarely occurs in meronym position in assumed meronymy relations, we approximate p by $P(me = t | arg2_r = a2)$, analogously for rarely occurring concepts in the holonym position. As usual, the probabilities are estimated by relative frequencies relying on a human annotation. Example: Let us consider that a hypothesis pair (c_1, c_2) is given and concept c_1 occurs in 90 annotated meronymy hypotheses as an assumed meronym, 45 of them are known to be correct. The concept c_2 occurs in 100 hypotheses as holonym candidate and 30 of them are known to be correct. Let the probability that a meronym hypothesis is correct be 0.2. Then the total score is given as: $(1/0.2) \cdot 0.5 \cdot 0.3 = 0.75$.

Graph Kernel: The use of kernels (see Section 2) is quite popular for semantic relation extraction. Since relation extraction algorithms are mainly based on syntactic or surface structures, tree or string kernels are usually applied. In our scenario, the kernel is only applied for validation of hypotheses where the extraction is done by the automated theorem prover. Hence, a hybrid approach is taken. In addition, our method is based on SNs, which are graphs and not trees. Thus, instead of the usual tree kernel, a graph kernel (vor der Brück and Helbig, 2010) based on common walks, as proposed by (Gärtner et al., 2003) is applied.

Applied Pattern: There are major differences regarding the precision values of the extracted meronymy hypotheses depending on the applied pattern (Berland and Charniak, 1999). Pattern D_1 (see Table 2) is actually quite reliable where D_2 generates a lot of incorrect hypotheses. Thus, we provide a feature for each pattern which takes the value of one if a meronymy hypothesis was extracted by this pattern, to zero otherwise.

Mutual Information: Relation hypotheses extracted several times are often more reliable than hypotheses that could only be found once. Thus, we introduce a feature measuring the point-wise mutual information (in contrast to the conditional probability in Berland and Charniak (1999)) between the meronym and holonym candidate multiplied by the discounting factor suggested in (Pantel and Ravichandran, 2004).

Deep/Shallow: This feature checks whether a hypothesis was extracted by both shallow and deep patterns (1) or only by one of them (0).

Concrete/Abstract: This feature follows the assumption that most meronymically related concepts are concrete. It is the product of the concreteness of both meronym and holonym candidates. The concreteness of a single concept is defined as the fraction of ontological sorts for a concept that are concrete (ontological sort: co or subordinated to co). Note that a concept can be assigned several ontological sorts if it is a meaning molecule (Helbig, 2006). A meaning molecule is assigned several meaning facets where each facet can have different ontological sorts, features and types of extensionality. Let us consider an example for the calculation of the concreteness. If a concept is assigned two concrete sorts and one abstract, then the concreteness of this concept is 2/3.

Ontological Sorts: Consider the case that at least one of the compared concepts involved in the meronymy hypothesis is a meaning molecule and is associated to several ontological sorts. The filtering process as described in Section 6.1 tests if there is at least one admissible combination of ontological sorts of the two concepts. All other ontological sorts are disregarded for this test. However, the disregarded ontological sorts can also give a clue about the hypothesis correctness. Therefore, we introduced a feature which employs the total set of ontological sorts of the two compared concepts.

A meronymy relation is presumably more likely to hold if the involved concepts are assigned similar sets of ontological sorts. Thus, this feature is set to the Jaccard coefficient considering the ontological sorts of the meronym (m) and holonym (h) candidates:

$$sort\_feature(m,h) := \frac{|sorts(m) \cap sorts(h)|}{|sorts(m) \cup sorts(h)|} \tag{13}$$

7 Meronymy Subrelations

The algorithm described thus far only extracts relations of type MERO. However, the meronymy relation is divided into several subrelations. In this section we describe how the correct subrelation is chosen. First, we introduce the set of all subrelations to choose from. Second the decision procedure is described in detail.

7.1 Types of Meronymy Relations

Winston proposes six subrelations for meronymy (Winston et al., 1987), which unfortunately are not sufficiently clearly defined. Therefore we base our approach on the meronymy relations of MultiNet (see (Helbig, 2006), Chapt. 4.2 and 18.2.49) which are systematically described and underpinned by an axiomatic apparatus. In the following it is tried to establish an approximate correspondence between both systems. Please note, that the division of meronymy of WordNet (Fellbaum, 1998) into three subrelations is considered by both aforementioned authors as being underspecified and not sufficiently differentiated. The following subrelations are proposed by Winston:

- Component-integral: A relation between an object and one of its components. Important for this relation is the fact that object and component can be perceived separately from each other. MultiNet relation: PARS (see Section 3). Example: *A car wheel is part of a car.*

- Member-collection: This relation represents the membership in a set. MultiNet relation: ELMT. Example: *A soccer player is a member of a soccer team.*
- Portion-mass: Relations which refer to mass units and their parts. MultiNet relation: PARS, for temporal units: TEMP. Example: *A meter is part of a kilometer, a slice of the cake is part of the cake.*
- Stuff-object: This relation represents the chemical composition of an object. MultiNet relation: ORIGM^{-1} if the holonym denotes a physical object, otherwise PARS. Example: *Alcohol is part of wine. Steel is part of a bike.*
- Feature-activity: Activities can usually be divided into several subactions. MultiNet relation: HSIT. Example: The following subactions belong to the activity *going out for dinner*: *visiting a restaurant*, *ordering*, *eating* and *payment*
- Place-area: This relation holds between two objects if one of these objects is geographically part of the other object. MultiNet relation: PARS. Example: *Germany is part of Europe.*

Additionally, (Helbig, 2006) defines a further meronymy subrelation for subsets called SUBM in MultiNet. Example: *A brigade is a subset of a division.* Note that brigade/division is not a member-collection relationship since both, a division and a brigade, denote concepts with sets as their extension, whose elements are soldiers.

Premise	Decision
$etype(m) + 1 = etype(h)$	ELMT
$sort(m) \sqsubseteq si \wedge$ $sort(h) \sqsubseteq si$	HSIT
$sort(m) \sqsubseteq s \wedge$ $sort(h) \sqsubseteq d$	ORIGM^{-1}
$etype(m) = etype(h) \wedge$ $etype(m) > 0$	SUBM
$sort(m) \sqsubseteq ta \wedge$ $sort(h) \sqsubseteq ta$	TEMP
otherwise	PARS

Table 5: Selecting the correct meronymy subrelation (d=discrete object, s=substance, si=situation, ta=temporal abstracta, m=meronym, h=holonym), $sort(x) \sqsubseteq y :\Leftrightarrow sort(x) = y$ or $sort(x)$ is a subsort of y.

7.2 Selecting the Correct Meronymy Subrelation

The semantic lexicon is employed to choose the correct subrelation for an extracted meronymy relation, i.e., this process is not based on any machine learning algorithm. The decision rules are given in Table 5. For the definition of ontological sorts and the type of extensionality, which are required for the decision process, see Section 3. This procedure requires that both concepts taken into consideration are contained in the lexicon. If this is not the case, then one of the following fall-back strategies are used. First, if one of the regarded concepts is represented by a compound word, as determined by a morphological compound analysis, then the lexical entry of the base concept (concept corresponding to the correct reading of the base word) can be used. Second, for correct meronymy relations, which is assumed for this selection, the semantic sorts of the two concepts must usually be identical (exception: ORIGM^{-1}). This means, for instance, that if the first concept is known to be of sort *ta* the second should have the same sort and the correct subrelation should be TEMP. If a concept is a meaning molecule, the facets are chosen from both concepts for comparison, which are most similar in regard to ontological sorts and semantic features.

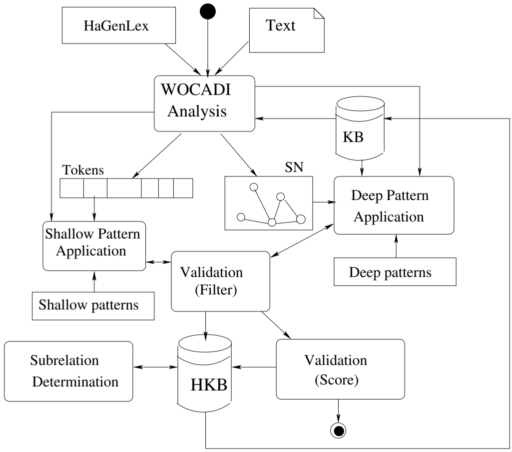

Figure 3: Activity diagram of the meronymy extraction done by SemQuire.

8 Architecture

To find meronymy relations from a text, this text is processed by our knowledge acquisition tool called SemQuire[8] (see Figure 3).

1. At first, the sentences of Wikipedia are analyzed by the deep linguistic parser WOCADI employing the knowledge base KB, containing the general background knowledge and the axiomatic apparatus.
   As a result of the parsing, a token list, a set of syntactic dependency trees, and a large semantic network (SN) are created.
2. Shallow patterns, consisting of a regular expression in the premise, are applied to the token lists, and deep patterns are applied to the SNs to generate proposals for meronymy relations (see Section 4 and Section 5).
3. A validation tool using ontological sorts and semantic features checks whether the proposals are at all technically admissible to reduce the amount of data stored in the hypotheses knowledge base HKB (see Section 6.1).
4. If the validation is successful, the meronymy candidate pair is added to the HKB. Steps 2–4 are repeated until all sentences are processed.
5. Each meronymy candidate pair in HKB is assigned a confidence score (see Section 6.2) estimating the likelihood of its correctness.
6. The correct meronymy subrelation is determined (see Section 7).
7. The highest scored hypotheses in HKB are manually inspected and eventually added to the knowledge base KB.

[8] SemQuire is derived from *acquire knowledge semantic-based.*

Feature	Correlation
Deep/Shallow	0.512
Ontological Sorts	0.436
Correctness Rate	0.397
Use of a Taxonomy	0.379
Concrete/Abstract	0.337
Mutual Information	0.030

Table 6: Correlation of features to relation correctness

9 Evaluation

The meronymy relations automatically acquired stem from the German Wikipedia corpus from November 2006 consisting of 500 000 articles and 20 million sentences.

In more than 6 million cases, a relation candidate was filtered out as being incorrect by our validation component. In total, 1 449 406 (different) meronymy relation hypotheses were finally stored in the knowledge base, 286 008 of them originating exclusively from deep patterns. In total, the relations in the knowledge base were extracted by approximately 2.5 million pattern applications.

	GermaNet		Costello		SemQuire		Sum
	PNM	PM	PNM	PM	PNM	PM	(PM+ PNM)
NM	750	0	506	244	666	84	750
M	718	32	393	357	151	599	750
Sum	1468	32	899	601	817	683	

Table 7: Confusion matrix for SVM optimization. NM = no meronymy relation present, M = meronymy relation present, PNM = predicted non-meronymy relation, PM = predicted meronymy relation.

For relation hypotheses extracted by deep and shallow patterns together, the precision is more than three times higher than for the relations that were only extracted by shallow patterns. 1 450 of the relations of the knowledge base were extracted alone by employing the SUB0 transitivity axiom A_2, exploiting the fact that a *district* de-

Measure	GermaNet	Costello	SemQuire
Accuracy	0.521	0.573	0.843
Recall	0.043	0.476	0.799
Precision	1.000	0.594	0.877
F-measure	0.082	0.528	0.837

Table 8: Accuracy, recall, precision, and F-measure for SVM optimization.

notes a *part* (see the example in Section 5). In total, logical axioms have been applied in 58 101 relation extraction processes, discovering 34 114 distinct relations. Table 4 shows a selection of axioms and the number of extracted hypotheses applying a certain axiom.

1 500 hypotheses were selected for the evaluation and annotated for correctness. Additional 50 000 hypotheses were annotated with their correctness exploited by the feature *Correctness Rate*, which is described in Section 6.2. Table 6 shows several scoring features and their associated correlation to the hypothesis correctness as specified by the annotators (one (1) for hypothesis is correct, zero (0) for incorrect).

Accuracy, precision, recall, F-measure, and confusion matrices were determined by a 10-fold cross-validation. Precision is the relative frequency with which a predicted meronym is actually one, while accuracy denotes the relative frequency with which the decision (meronymy/non-meronymy) is in fact correct. Note that recall does not relate to the extraction process, but rather only to the (score-based) validation. Thus, it specifies the relative frequency with which a correct relation in the data set of 1500 relations is actually identified as being correct by our system.

Our approach is compared with a GermaNet classifier as a baseline that predicts a relation of our hypotheses set to be meronymic if this relation is contained in GermaNet (GermaNet synsets are mapped to HaGenLex concepts semi-automatically) or can be derived by other GermaNet relations. A second baseline is the validation feature of Costello (Costello, 2007) which is also used as a feature by our system. The evaluation results are given in Tables 7 and 8. The evaluation showed that we were able to find a lot of meronyms not contained in GermaNet. In particular, less than 6% of the meronyms identified by SemQuire were contained or derivable from GermaNet. In addition, our extracted meronymy relations are, in contrast to GermaNet, all concept-based and not synset-based.[9]

They are also further differentiated into several subrelations. The correct subrelation was determined in 92.5% of the cases.

The results of our system are quite competitive in comparison with the results obtained by Girju et al. (2006) (precision: 0.81 and recall: 0.759). For this comparison, one has to take into consideration that Girju's approach is operated with English data and employs semantic relations from WordNet. Therefore, for his results, larger lexical resources were available than for German, which makes this task for a German text corpus more difficult.

The runtime of the algorithm heavily depends on the theorem prover timeout, i.e., the maximum amount of time which is available for a single proof. Currently the timeout is set to 0.1 seconds. With that the total runtime of the meronymy extraction algorithm is about three weeks on a Intel Core 2 Quad Q9550 CPU with 2.83 GHz using 8MiB of memory. By reducing the timeout the calculation time can be arbitrarily

[9] The difference is seen in the fact that MultiNet concepts are embedded in a complex linguistic and logical apparatus. Thus, concept ids of MultiNet are present in meaning postulates and other logical axioms, they are contained in the analysis results derived automatically by word disambiguation from the lexicon, and so on. This embedding in a whole process of language understanding is lacking in the case of representatives of synsets.

reduced. Naturally, the number of extracted hypotheses will decrease with descending timeout threshold. The realistic lower limit for the entire processing time, where still a reasonable amount of hypotheses can be found, is two days.

10 Conclusion and Future Work

In this paper, a logic-oriented approach has been presented for extracting meronymy relations from Wikipedia via text mining, which has proven its value in acquiring large stocks of knowledge. Unlike other approaches, our methods are based on a deep semantic representation, employing logical axioms. The use of axioms improves the generality of the method and can therefore increase the recall of the patterns in terms of the number of extracted meronymy relations. Furthermore, axioms can also be used to improve the precision of the patterns.

For future work, we plan on increasing the number of axioms and transferring this approach to other semantic relations.

The application of semantic patterns and axioms proved to be important for meronymy detection and is, in our opinion, an important step towards the use of real text understanding for future knowledge extraction systems.

Acknowledgement

We wish to thank all members of our department for their support. Especially we wish to thank Tiansi Dong for proof-reading this paper. This work is partly funded by the DFG project *Semantische Duplikatserkennung mithilfe von Textual Entailment* (HE 2847/11-1).

References

Baumgartner, P., Furbach, U., and Pelzer, B. (2007). Hyper tableaux with equality. In *Automated Deduction – CADE-21*, volume 4603 of *LNCS*, pages 492–507. Springer, Heidelberg, Germany.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 57–64, College Park, Maryland.

Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 724–731, Vancouver, Canada.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.* Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Costello, F. J. (2007). UCD-FC: Deducing semantic relations using WordNet senses that occur frequently in a database of noun-noun compounds. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, pages 370–373, Prague, Czech Republic.

Culotta, A. and Sorenson, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–429, Barcelona, Spain.

Fellbaum, C., editor (1998). *WordNet An Electronic Lexical Database.* MIT Press, Cambridge, Massachusetts.

Friedmann, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, pages 129–143, Washington, District of Columbia.

Girju, R., Badulescu, A., and Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Glöckner, I. (2007). Answer validation through robust logical inference. In Peters, C. et al., editors, *Evaluation of Multilingual and Multi-modal Information Retrieval: 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006*, number 4730 in Lecture Notes in Computer Science (LNCS), pages 518–521. Springer, Heidelberg, Germany.

Hamp, B. and Feldweg, H. (1997). GermaNet - a lexical-semantic net for german. In *Proceedings of the ACL/EACL workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

Hartrumpf, S. (2002). *Hybrid Disambiguation in Natural Language Analysis.* PhD thesis, FernUniversität in Hagen, Fachbereich Informatik, Hagen, Germany.

Hartrumpf, S., Helbig, H., and Osswald, R. (2003). The semantically based computer lexicon HaGenLex – Structure and technological environment. *Traitement automatique des langues*, 44(2):81–105.

Helbig, H. (2006). *Knowledge Representation and the Semantics of Natural Language.* Springer, Heidelberg, Germany.

Lyons, J. (2002). *Linguistic Semantics - An introduction.* Cambridge University Press, Cambridge, UK.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the Conference on Computational Linguistics/Annual Meeting of the Association of Computational Linguistics (COLING/ACL)*, pages 113–120.

Pantel, P. and Ravichandran, D. (2004). Automatically labeling semantic classes. In *Proceedings of the Human Language Technology Conference and the Conference of the North American chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL)*, pages 321–328, Boston, Massachusetts.

Reichartz, F., Korte, H., and Paass, G. (2009). Dependency tree kernels for relation extraction from natural language text. In *Proceedings of the European Conference on Machine Learning and the Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 270–285, Bled, Slovenia.

Suchanek, F. N., Sozio, M., and Weikum, G. (2009). SOFIE: a self-organizing framework for information extraction. In *Proceedings of the 18th International Conference on the World Wide Web (WWW)*, pages 631–640.

vor der Brück, T. and Helbig, H. (2010). Validating meronymy hypotheses with support vector machines and graph kernels. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 243–250, Washington, District of Columbia.

vor der Brück, T. and Stenzhorn, H. (2010). Logical ontology validation using an automatic theorem prover. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI)*, pages 491–496, Lisbon, Portugal.

Winston, M. E. et al. (1987). A taxonomy of part-whole relations. *Cognitive Science*, 11(4):417–444.

Zhao, S. and Grishman, R. (2005). Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426.