Discovering Conceptual Relations from Text

A. Maedche and S. Staab
{maedche, staab}@aifb.uni-karlsruhe.de
Institute AIFB, Karlsruhe University, Germany
http://www.aifb.uni-karlsruhe.de/WBS
Tel.: +49-721-608 6558   Fax: +49-721-693717

Abstract

Ontologies have become an important means for structuring information and information systems and, hence, important in knowledge as well as in software engineering. However, there remains the problem of engineering large and adequate ontologies within short time frames in order to keep costs low. For this purpose, efforts have been made to facilitate the ontology engineering process, in particular the acquisition of ontologies from domain texts. We broaden these investigations with regard to two dimensions. First, we present a general architecture for discovering ontological concepts and relations. This architecture is general enough to subsume current approaches in this direction. Second, we propose a new approach to extend current approaches, who mostly focus on the semi-automatic acquisition of taxonomies, by the discovery of non-taxonomic conceptual relations. We use a generalized association rule algorithm that does not only detect relations between concepts, but also determines the *appropriate level of abstraction* at which to define relations. This is crucial for an appropriate ontology definition in order that it be succinct and conceptually adequate and, hence, easy to understand, maintain, and extend. In order to prove the validity of our proposal we evaluate the success of our learning approach against a manually engineered ontology. For this objective, we present a new paradigm suited to evaluate the degree to which relations that are learned match relations in a manually engineered ontology.

1 Introduction

Ontologies[1] have shown their usefulness in application areas such as intelligent information integration or information brokering by providing a technical means to share and exchange knowledge and/or information between humans and/or machines (Wiederhold, 1993; Abecker et al., 1999; Schnurr & Staab, 2000). Hence, their importance for software and knowledge engineering may hardly be overestimated. Nevertheless, their wide-spread usage is still hindered by ontology engineering being rather time-consuming and, hence, expensive. Therefore a number of proposals have been made to facilitate ontology engineering through automatic discovery from domain data,

[1] We restrict our attention in this paper to *domain ontologies* that describe a particular small model of of the world as relevant to applications, in contrast to *top-level ontologies* and *representational ontologies* that aim at the description of generally applicable conceptual structures and meta-structures, respectively, and that are mostly based on philosophical and logical point of views rather than focused on applications.

domain-specific natural language texts in particular (cf. (Byrd & Ravin, 1999; Faure & Nedellec, 1998; Hahn & Schnattinger, 1998; Morin, 1999; Resnik, 1993; Wiemer-Hastings et al., 1998)). However, we see two pitfalls occur in most of these seminal approaches.

First, these investigation have mostly been conceived in isolation from actual issues of ontology engineering systems. A framework for classification and evaluation of approaches is lacking. Thus, the overall picture of what resources may or should be used in ontology discovery approaches remains rather vague and has not been under discussion at all.

Second, most of these approaches have only looked at how to learn the taxonomic part of ontologies. In applications like (Wiederhold, 1993; Abecker et al., 1999; Schnurr & Staab, 2000), an ontology O often boils down to a an object model represented by a set of concepts C, which are *taxonomically* related by the transitive ISA relation $H \subset C \times C$ and *non-taxonomically* related by named object relations $R^* \subset C \times C \times \texttt{String}$. On the basis of the object model a set of logical axioms, A, enforce semantic constraints. Common approaches mostly focus on the automatic acquisition of C and H and often neglect the importance of interlinkage between concepts. Though taxonomic knowledge is certainly of utmost importance, major efforts in ontology engineering must be dedicated to the definition of *non-taxonomic conceptual relationships*, e.g. hasPart relations between concepts. The determination of non-taxonomic conceptual relationships is not this well-researched.[2] In fact, it appears to be the more intricate task as, in general, it is less well known how many and what type of conceptual relationships should be modeled in a particular ontology.

This paper presents a framework for semi-automatic engineering of ontologies. Within our general architecture (Section 2), we embed a new approach for discovering non-taxonomic conceptual relations from text and, hence, for facilitating the engineering of non-taxonomic relations. Building on the taxonomic part of the ontology, our approach analyzes domain-specific texts. It uses shallow text processing methods to identify linguistically related pairs of words (cf. Section 3). An algorithm for discovering generalized association rules analyzes statistical information about the linguistic output (cf. Section 4). Thereby, it uses the background knowledge from the taxonomy in order to propose relations at the appropriate level of abstraction. For instance, the linguistic processing may find that the word "costs" frequently co-occurs with each of the words "hotel", "guest house", and "youth hostel" in sentences such as (1).[3]

(1) Costs at the youth hostel amount to \$ 20 per night.

From this statistical linguistic data our approach derives correlations at the conceptual level, viz. between the concept Costs and the concepts, Hotel, Guest House, and Youth Hostel. The discovery algorithm determines support and confidence measures for the relationships between these three pairs, as well as for relationships at higher levels of abstraction, such as between Accommodation and Costs. In a final step, the algorithm determines the level of abstraction most suited to describe the conceptual relationships by pruning appearingly less adequate ones. Here, the relation between Accommodation and Costs may be proposed for inclusion in the ontology. A more comprehensive example will be presented in Section 5.

[2] An informal survey performed by a former colleague found that a number of prominent and freely available ontologies, like WordNet or Sensus, lacked rich interlinking of concepts through conceptual relations.

[3] For ease of presentation we mostly give English examples, however, our evaluation is based on our implementation that processes German texts.

Finally, we also evaluate our approach against an ontology about the tourism domain that we had modeled before using standard knowledge engineering techniques. Linguistic processing was done on a text corpus extracted from a web site about tourist information. We have performed evaluation with regard to standard measures, however, we have also found that evaluation needs to take account of the sliding scale of adequacy prevalent in a hierarchical target structure. Thus, we have also conceived of a new evaluation measure to evaluate our experiments (cf. Section 6). We conclude with a survey of related work and a short remark on the acquisition of ontological axioms, A.

2 An Architecture for Semi-Automatic Ontology Acquisition

The purpose of this section is to give an overview of the architecture of our system Text-To-Onto (cf. the overall schema in Figure 1 and the snapshot in Figure 2). The process of semi-automatic ontology acquisition is embedded in an application that comprises several core features described as a kind of pipeline in the following. Nevertheless, the reader may bear in mind that the overall development of ontologies remains a cyclic process (cf. (Maedche et al., 2000)). In fact, we provide a broad set of interactions such that the engineer may start with primitive methods first. These methods require very little or even no background knowledge, but they may also be restricted to return only simple hints, like term frequencies. While the knowledge model matures during the semi-automatic engineering process, the engineer may turn towards more advanced and more knowledge-intensive algorithms, such as our mechanism for discovering generalized relations.

Text & Processing Management Component. The ontology engineer uses the Text & Processing Management Component to select domain texts exploited in the further discovery process. She chooses among a set of text (pre-)processing methods available on the Text Processing Server and among a set of algorithms available at the Learning & Discovering component. The former module returns text that is annotated by XML and this XML-tagged text is fed to the Learning & Discovering component.

Text Processing Server. The Text Processing Server may comprise a broad set of different methods. In our case, it contains a shallow text processor based on the core system SMES (Saarbrücken Message Extraction System). SMES is a system that performs syntactic analysis on natural language documents. Its functionality is described in detail in Section 3. In general, the Text Processing Server is organized in modules, such as a tokenizer, morphological and lexical processing, and chunk parsing that use lexical resources to produce mixed syntactic/semantic information. The results of text processing are stored in annotations using XML-tagged text.

Lexical DB & Domain Lexicon. Syntactic processing relies on lexical knowledge. In our system, SMES accesses a lexical database with more than 120.000 stem entries and more than 12,000 subcategorization frames that are used for lexical analysis and chunk parsing. The domain-specific part of the lexicon (abbreviated “domain lexicon”; cf. left upper part of Figure 2) associates word

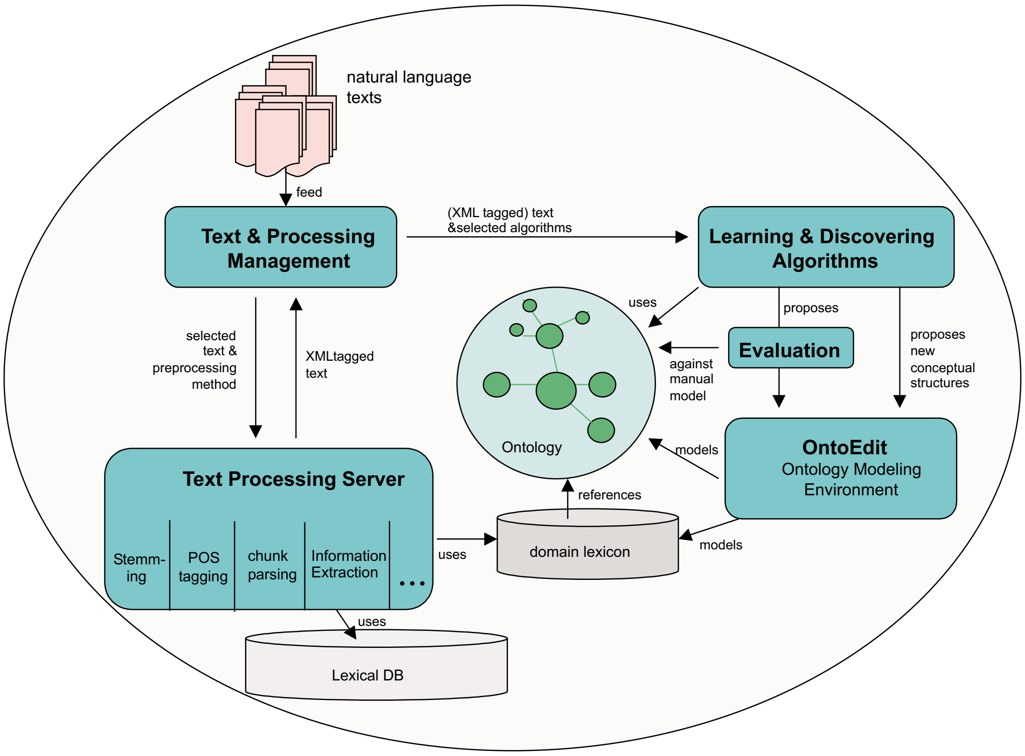


Figure 1: Architecture of the Ontology Learning Environment

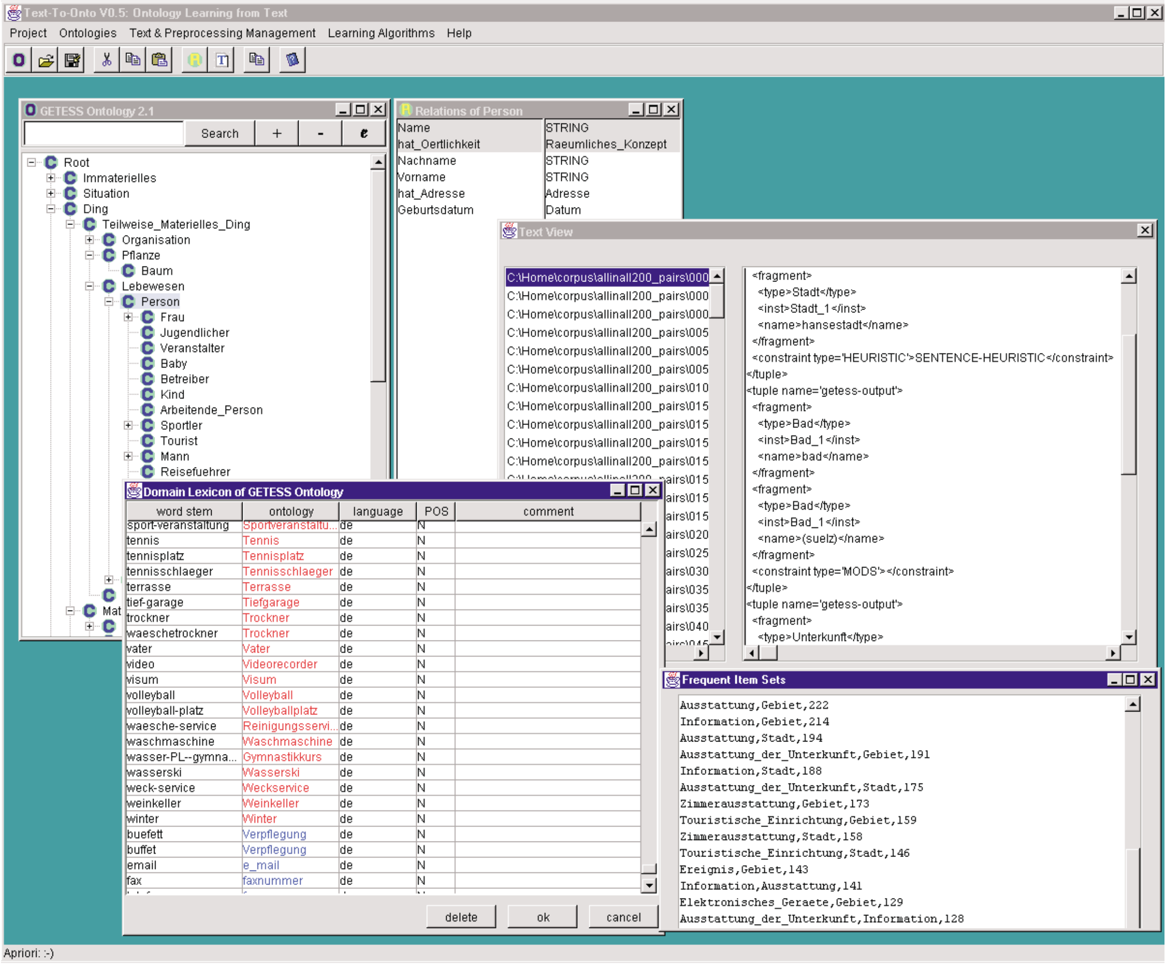
stems with concepts available in the concept taxonomy. Hence, it links syntactic information with semantic knowledge that may be further refined in the ontology.

Learning & Discovering component. The Learning & Discovering component uses various discovering methods on the annotated texts, e.g. term extraction methods for concept acquisition. Our scenario for discovering non-taxonomic relations uses the learning algorithm for discovering generalized association rules described in Section 4. Conceptual structures that exist at learning time (e.g. a concept taxonomy) may be incorporated into the learning algorithms as background knowledge. The evaluation such as described in Section 6 is performed in a submodule based on the results of the learning algorithm.

Ontology Modeling Environment. The Ontology Modeling Environment (**OntoEdit**[4]) supports the ontology engineer in semi-automatically adding newly discovered conceptual structures to the ontology.[5] The screenshot depicted in Figure 2 shows on the left side the object-model backbone of an ontology, i.e. the sets C, H, and R^*. In addition to core capabilities for structuring

[4] OntoEdit is a submodule of the Ontology Learning Environment “Text-To-Onto”.

[5] A comprehensive description of the ontology engineering system OntoEdit and the underlying methodology is given in (Staab & Maedche, 2000).



```
Root
  Immaterielles
  Situation
  Ding
    Teilweise_Materielles_Ding
      Organisation
      Pflanze
        Baum
      Lebewesen
        Person
          Frau
          Jugendlicher
          Veranstalter
          Baby
          Betreiber
          Kind
          Arbeitende_Person
          Sportler
          Tourist
          Mann
          Reisefuehrer
    Mat
```

Name	STRING
hat_Oertlichkeit	Raeumliches_Konzept
Nachname	STRING
Vorname	STRING
hat_Adresse	Adresse
Geburtsdatum	Datum

```
C:\Home\corpus\allinall200_pairs\000
C:\Home\corpus\allinall200_pairs\000
C:\Home\corpus\allinall200_pairs\000
C:\Home\corpus\allinall200_pairs\005
C:\Home\corpus\allinall200_pairs\005
C:\Home\corpus\allinall200_pairs\005
C:\Home\corpus\allinall200_pairs\010
C:\Home\corpus\allinall200_pairs\015
C:\Home\corpus\allinall200_pairs\015
C:\Home\corpus\allinall200_pairs\015
C:\Home\corpus\allinall200_pairs\015
C:\Home\corpus\allinall200_pairs\015
airs\015
airs\015
airs\020
airs\025
airs\030
airs\035
airs\035
airs\040
airs\045
```

```
 <fragment>
  <type>Stadt</type>
  <inst>Stadt_1</inst>
  <name>hansestadt</name>
 </fragment>
 <constraint type='HEURISTIC'>SENTENCE-HEURISTIC</constraint>
</tuple>
<tuple name='getess-output'>
 <fragment>
  <type>Bad</type>
  <inst>Bad_1</inst>
  <name>bad</name>
 </fragment>
 <fragment>
  <type>Bad</type>
  <inst>Bad_1</inst>
  <name>(suelz)</name>
 </fragment>
 <constraint type='MODS'></constraint>
</tuple>
<tuple name='getess-output'>
 <fragment>
  <type>Unterkunft</type>
```

word stem	ontology	language	POS	comment
sport-veranstaltung	Sportveranstaltu...	de	N	
tennis	Tennis	de	N	
tennisplatz	Tennisplatz	de	N	
tennisschlaeger	Tennisschlaeger	de	N	
terrasse	Terrasse	de	N	
tief-garage	Tiefgarage	de	N	
trockner	Trockner	de	N	
waeschetrockner	Trockner	de	N	
vater	Vater	de	N	
video	Videorecorder	de	N	
visum	Visum	de	N	
volleyball	Volleyball	de	N	
volleyball-platz	Volleyballplatz	de	N	
waesche-service	Reinigungsservi...	de	N	
waschmaschine	Waschmaschine	de	N	
wasser-PL--gymna...	Gymnastikkurs	de	N	
wasserski	Wasserski	de	N	
weck-service	Weckservice	de	N	
weinkeller	Weinkeller	de	N	
winter	Winter	de	N	
buefett	Verpflegung	de	N	
buffet	Verpflegung	de	N	
email	e_mail	de	N	
fax	faxnummer	de	N	

```
Ausstattung,Gebiet,222
Information,Gebiet,214
Ausstattung,Stadt,194
Ausstattung_der_Unterkunft,Gebiet,191
Information,Stadt,188
Ausstattung_der_Unterkunft,Stadt,175
Zimmerausstattung,Gebiet,173
Touristische_Einrichtung,Gebiet,159
Zimmerausstattung,Stadt,158
Touristische_Einrichtung,Stadt,146
Ereignis,Gebiet,143
Information,Ausstattung,141
Elektronisches_Geraete,Gebiet,129
Ausstattung_der_Unterkunft,Information,128
```

Figure 2: The Text-To-Onto Ontology Learning Environment

the ontology, the engineering environment provides some additional features for the purpose of documentation, maintenance, and ontology exchange.

3 Shallow Text Processing

Our approach has been implemented on top of SMES (Saarbrücken Message Extraction System), a shallow text processor for German (cf. (Neumann et al., 1997)) that has been adapted to the tourism domain. This is a generic component that adheres to several principles that are crucial for our objectives. *(i)*, it is fast fast and robust, *(ii)*, it yields dependency relations between terms, and, *(iii)*, it returns pairs of concepts the coupling of which is motivated through *linguistic* constraints on the corresponding textual terms. In addition, we made some minor changes such that principle *(iv)*, linguistic processing delivers a high recall on the number of dependency relations occuring in a text, is also guaranteed. We here give a short survey on SMES in order provide the reader with a comprehensive picture of what underlies our evaluation.

The **Architecture** of our Text Processing Server, SMES, comprises a *tokenizer* based on regular expressions, a *lexical analysis* component, and a *chunk parser*.

Tokenizer. Its main task is to scan the text in order to identify boundaries of words and complex expressions like “$20.00” or “Mecklenburg-Vorpommern’[6], and to expand abbreviations.

Lexical Analysis uses lexical information to perform, *(1)*, morphological analysis, *i.e.*, the identification of the canonical common stem of a set of related word forms and the analysis of compounds, *(2)*, recognition of name entities, *(3)*, retrieval of domain-specific information, and, *(4)*, part-of-speech tagging:

1. In German compounds are extremely frequent and, hence, their analysis into their parts, e.g. “database” becoming “data” and “base”, is crucial and may yield interesting relationships between concepts. Furthermore, morphological analysis returns possible readings for the words concerned, e.g. the noun and the verb reading for a word like “man” in “The old man the boats.”

2. Processing of named entities includes the recognition of proper and company names like “Hotel Schwarzer Adler” as single, complex entities, as well as the recognition and transformation of complex time and date expressions into a canonical format, e.g. “January 1st, 2000” becomes “1/1/2000”.

3. The next step associates single words or complex expressions with a concept from the ontology if a corresponding entry in the domain-specific part of the lexicon exists. E.g., the expression “Hotel Schwarzer Adler” is associated with the concept Hotel.

4. Finally, part-of-speech tagging disambiguates the reading returned from morphological analysis of words or complex expressions using the local context.

Chunk Parser. SMES uses weighted finite state transducers to efficiently process phrasal and sentential patterns. The parser works on the phrasal level, before it analyzes the overall sentence. Grammatical functions (such as subject, direct-object) are determined for each dependency-based sentential structure on the basis of subcategorizations frames in the lexicon.

Dependency Relations. Our primary output derived from SMES consists of *dependency relations* (Hudson, 1990) found through lexical analysis (compound processing) and through parsing

[6] Mecklenburg-Vorpommern is a region in the north east of Germany.

at the phrase and sentential level. It is important for our approach that on these levels syntactic dependency relations coincide rather closely with semantic relations that are often found to hold between the very same entities (cf. (Hajicova, 1987)). Thus, we derived our motivation to output those conceptual pairs to the learning algorithm the corresponding terms of which are dependentially related. Thereby, the grammatical dependency relation need not even hold directly between two conceptually meaningful entities. For instance, in (2) "Hotel Schwarzer Adler" and "Rostock", the concepts of which appear in the ontology as Hotel and City, respectively, are not directly connected by a dependency relation. However, the preposition "in" acts as a mediator that incurs the conceptual pairing of Hotel with City (cf. (Romacker et al., 1999) for a complete survey of mediated conceptual relationships).

(2) The *Hotel Schwarzer Adler* in *Rostock* celebrates Christmas.

Heuristics. Chunk parsing such as performed by SMES still returns many phrasal entities that are not related within or across sentence boundaries. This however means that our approach would be doomed to miss many relations that often occur in the corpus, but that may not be detected due to the limited capabilities of SMES. For instance, it does not attach prepositional phrases in any way and it does not handle anaphora, to name but two desiderata. We have decided that we needed a high recall of the linguistic dependency relations involved, even if that would incur a loss of linguistic precision. The motivation is that with a low recall of dependency relations the subsequent algorithm may learn only very little, while with less precision the learning algorithm may still sort out part of the noise. Therefore, the SMES output has been extended to include heuristic correlations beside linguistics-based dependency relations:

- The *NP-PP-heuristic* attaches all prepositional phrases to adjacent noun phrases.
- The *sentence-heuristic* relates all concepts contained in one sentence if other criteria fail. This is a crude heuristic that needs further refinement. However, we found that it yielded many interesting relations, e.g. for enumerations, which could not be parsed successfully.
- The *title-heuristic* is very specific for our domain. It links the concepts such as referred to in the HTML title tags with all the concepts contained in the the overall document. This strategy might utterly fail in other domains, but it was successful for our hotel and sight descriptions.

To sum up, linguistic processing outputs a set of concept pairs, $CP := \{(a_{i,1}, a_{i,2}) | a_{i,j} \in C\}$. Their coupling is motivated through various direct and mediated linguistic constraints or by several general or domain-specific heuristic strategies.

4 Learning Algorithm

Our learning mechanism is based on the algorithm for discovering generalized association rules proposed by Srikant and Agrawal (Srikant & Agrawal, 1995). Their algorithm finds associations that occur between items, e.g. supermarket products, in a set of transactions, e.g. customers' purchases, and describes them at the appropriate level of abstraction, e.g. "snacks are purchased together with drinks" rather than "chips are purchased with beer" and "peanuts are purchased with soda".

The basic association rule algorithm is provided with a set of transactions $T := \{t_i | i = 1 \ldots n\}$, where each transaction t_i consists of a set of items $t_i := \{a_{i,j} | j = 1 \ldots m_i, a_{i,j} \in C\}$ and each item $a_{i,j}$ is from a set of concepts C. The algorithm computes *association rules* $X_k \Rightarrow Y_k$ $(X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ such that measures for *support* and *confidence* exceed user-defined thresholds. Thereby, support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset, and confidence for $X_k \Rightarrow Y_k$ is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction, *viz.*

(3) $\text{support}(X_k \Rightarrow Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{n}$

(4) $\text{confidence}(X_k \Rightarrow Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{|\{t_i | X_k \subseteq t_i\}|}$

Srikant and Agrawal have extended this basic mechanism to determine associations at the right level of a *taxonomy*, formally given by a taxonomic relation $H \subset C \times C$. For this purpose, they first extend each transaction t_i to also include each ancestor of a particular item $a_{i,j}$, i.e. $t'_i := t_i \cup \{a_{i,l} | (a_{i,j}, a_{i,l}) \in H\}$. Then, they compute confidence and support for all possible association rules $X_k \Rightarrow Y_k$ where Y_k does not contain an ancestor of X_k as this would be a trivially valid association. Finally, they prune all those association rules $X_k \Rightarrow Y_k$ that are subsumed by an "ancestral" rule $\hat{X}_k \Rightarrow \hat{Y}_k$, the itemsets $\hat{X}_k, \hat{Y}_k$ of which only contain ancestors or identical items of their corresponding itemset in $X_k \Rightarrow Y_k$.

For the discovery of conceptual relations we may directly build on their scheme, as described in the following four steps that summarize our learning module:

1. Determine $T := \{\{a_{i,1}, a_{i,2}, \ldots, a_{i,m'_i}\} | (a_{i,1}, a_{i,2}) \in CP \wedge$
   $l \geq 3 \rightarrow ((a_{i,1}, a_{i,l}) \in H \vee (a_{i,2}, a_{i,l}) \in H)\}$.

2. Determine support for all association rules $X_k \Rightarrow Y_k$, where $|X_k| = |Y_k| = 1$.

3. Determine confidence for all association rules $X_k \Rightarrow Y_k$ that exceed user-defined support in step 2.

4. Output association rules that exceed user-defined confidence in step 3 and that are not pruned by ancestral rules with higher or equal confidence and support.

Thus, the output of association rules are pairs of concepts that are proposed to the engineer for inclusion in the ontology as non-taxonomic relations $D := \{d_i\}$. The reader may note two important observations here.

First, we abstract from the naming of relations in our approach. Though this may certainly lead to unwanted conflations of relations, like (Person,Person,HIT) with (Person,Person,LOVE), we consider this a secondary concern for our interactive approach — though, of course, this is a major issue for further research.

Second, we here have chosen a baseline approach considering the determination of the set of transactions T. Actually, one may conceive of many strategies that cluster multiple concept pairs into one transaction. For instance, given a set of 100 texts each describing a particular hotel in detail. Each hotel might come with an address, but it might also have an elaborate description of the different types of public and private rooms and their furnishing resulting in 10,000 concept pairs returned from linguistic processing. Our baseline choice considers each concept pair as a transaction. Then support for the rule $\{\textsf{Hotel}\}\Rightarrow\{\textsf{Address}\}$ is equal or, much more probably, (far) less than 1%, while rules about rooms and their furnishing or their style, like $\{\textsf{Room}\}\Rightarrow\{\textsf{Bed}\}$, might achieve ratings of several percentage points. This means that an important relationship between $\{\textsf{Hotel}\}$ and $\{\textsf{Address}\}$ might get lost among other conceptual relationships. In contrast, if one considers complete texts to constitute transactions, an ideal linguistic processor might lead to more balanced support measures for $\{\textsf{Hotel}\}\Rightarrow\{\textsf{Address}\}$ and $\{\textsf{Room}\}\Rightarrow\{\textsf{Bed}\}$ of up to 100% each.

Thus, discovery might benefit when background knowledge about the domain texts is exploited for compiling transactions. In the future, we will have to further investigate the effects of different strategies.

5 Example

For the purpose of illustration, this chapter gives a comprehensive example, which is based on our actual experiments. We have processed a text corpus by a WWW provider for tourist information (URL: http://www.all-in-all.de). The corpus describes actual objects, like locations, accomodations, furnishings of accomodations, administrative information, or cultural events, such as given in the following example sentences.

(5) a. *Mecklenburg's* schönstes *Hotel* liegt in Rostock. (*Mecklenburg's* most beautiful *hotel* is located in Rostock.)
b. Ein besonderer Service für unsere Gäste ist der Frisörsalon in unserem Hotel. (A *hairdresser* in our *hotel* is a special service for our guests.)
c. Das Hotel Mercure hat *Balkone* mit direktem *Strandzugang*. (The hotel Mercure offers *balconies* with direct *access* to the beach.)
d. Alle *Zimmer* sind mit *TV*, Telefon, Modem und Minibar ausgestattet. (All *rooms* have *TV*, telephone, modem and minibar.)

Processing the example sentences (5a) and (5b), SMES (Section 3) outputs dependency relations between the terms, which are indicated in *slanted fonts* (and some more). In sentences (5c) and (5d) the heuristic for prepositional phrase-attachment and the sentence heuristic relate pairs of terms (marked by *slanted fonts*), respectively. Thus, four concept pairs – among many others – are derived with knowledge from the domain lexicon (cf. Table 1).

Table 1: Examples for linguistically related pairs of concepts

Term$_1$	$a_{i,1}$	Term$_2$	$a_{i,2}$
Mecklenburgs	area	*hotel*	hotel
hairdresser	hairdresser	*hotel*	hotel
balconies	balcony	*access*	access
room	room	*TV*	television

The algorithm for learning generalized association rules (cf. Section 4) uses the domain taxonomy, an excerpt of which is depicted in Figure 3, and the concept pairs from above (among many other concept pairs). In our actual experiments, we have defined a set of 284 concepts, $C := \{c_i\}$, and the domain-specific part of the lexicon has contained 486 entries referring to one of these concepts.

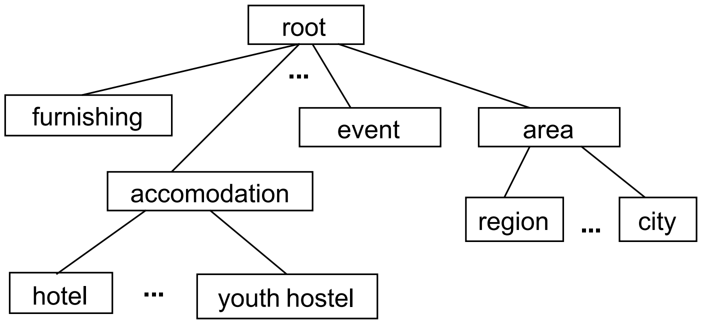


Figure 3: An example scenario

The learning algorithm discovered a large number of interesting and important non-taxonomic conceptual relations. A few of them are listed in Table 2. Note that in this table we also list two conceptual pairs, viz. (area, hotel) and (room, television), that are not presented to the user, but that are pruned. The reason is that there are ancestral association rules, viz. (area, accomodation) and (room, furnishing), respectively with higher confidence and support measures.

Table 2: Examples of discovered relations

Discovered relation	Confidence	Support
(area, accomodation)	0.38	0.04
~~(area, hotel)~~	~~0.1~~	~~0.03~~
(room, furnishing)	0.39	0.03
~~(room, television)~~	~~0.29~~	~~0.02~~
(accomodation, address)	0.34	0.05
(restaurant, accomodation)	0.33	0.02

6 Evaluation

For our evaluation we analyzed 2234 HTML documents, 16 million words and HTML tags, from our text corpus (cf. Section 5) with SMES (Section 3). The linguistic and heuristic preprocessing came up with approx. 51,000 linguistically related pairs, such as the ones in Table 1. For our overall project we had modeled an ontology, which contained 284 concepts and 88 non-taxonomic conceptual relations. The ontology, $O := (C, H, R)$, served for two purposes. On the one hand, the taxonomic structure of concepts, C, of our domain ontology was given as an input, viz. as the taxonomic relation $H \subset C \times C$, to the learning algorithm described in Section 4. On the other hand, we evaluated the success of our learning approach against the set of unnamed, non-taxonomic relations, $R \subset C \times C$, that had been hand-coded into the very same ontology before. Thus, we could compare the learning approach against human performance. Though human decisions in this matter should not be taken for pure gold[7], we think it is necessary to have measures that allow the comparison of different approaches and parameter settings — even when the bases of these measures depend to some extent on the quality of and on rather arbitrary, but equally plausible, choices between modeling decisions.

Precision and Recall. The first measures that we considered were *precision* and *recall* such as often used in information retrieval. When we denote the set of discovered relations by $D \subset C \times C$, they are defined by precision := $|D \cap R|/|D|$ and recall := $|D \cap R|/|R|$.

Running our experiments we found that precision and recall gave us some hints about how to gauge our thresholds for support and confidence (cf. Table 3). Nevertheless, these measures lacked a sense for the sliding scale of adequacy prevalent in our hierarchical target structures. To evaluate the quality of relations proposed to the ontology engineer, we also wanted to add some bonus to relations that *almost* fitted a hand-coded relation and, then, to compare different learning schemes on this basis. For this reason, we conceived of a new evaluation measure that reflected the distance between the automatically discovered relations D and the set of non-taxonomic, hand-coded relations R.

Relation Learning Accuracy ($\overline{\textbf{RLA}}$) is defined to capture intuitive notions for relation matches like “utterly wrong”, “rather bad”, “near miss” and “direct hit”. $\overline{\text{RLA}}$ is the averaged accuracy that the instances d of discovered relations D match against their best counterparts from R — disregarding arbitrary relational directions.

(6) $\overline{\text{RLA}}(D,R) = \frac{1}{|D|}\sum_{d \in D} \text{RLA}(d,R)$.

(7) $\text{RLA}(d,R) = \max_{r \in R, R^{-1}}\{\text{MA}(d,r)\}$.

We determine the accuracy that two relations match, MA, based on the geometric mean value of how close their domain and range concepts match such as given by the conceptual learning accuracy CLA (Note that $\text{MA}(d,r) = \text{MA}((a_1,a_2),(b_1,b_2))$).[8]

(8) $\text{MA}((a_1,a_2),(b_1,b_2)) := \sqrt{\text{CLA}(a_1,b_1) \cdot \text{CLA}(a_2,b_2)}$.

CLA is very similar in style to learning accuracy as introduced by Hahn & Schnattinger (Hahn & Schnattinger, 1998) who evaluate the categorization of unknown objects in a taxonomy. However, they assume that the target concept that is to be learned is always a leaf concept and, hence, a categorization learned for an object may not be more specific than the correct categorization. In our approach this assumption does not hold, hence our CLA differs from their measure. Basically, this accuracy measure reaches 100% when both concepts coincide (i.e., their distance $\delta(a,b)$ in the taxonomy H is 0); it degrades to the extent to which their distance increases; however, this degra-

[7] In fact, we are currently preparing an experiment. We want to determine the extent to which conceptual relations coincide when several ontology engineers introduce them independently from each other into a given taxonomy.

[8] The geometric mean reflects the intuition that if either domain or range concepts utterly fail to match, the matching accuracy converges against 0, whereas the arithmetic mean value might still turn out a value of 0.5.

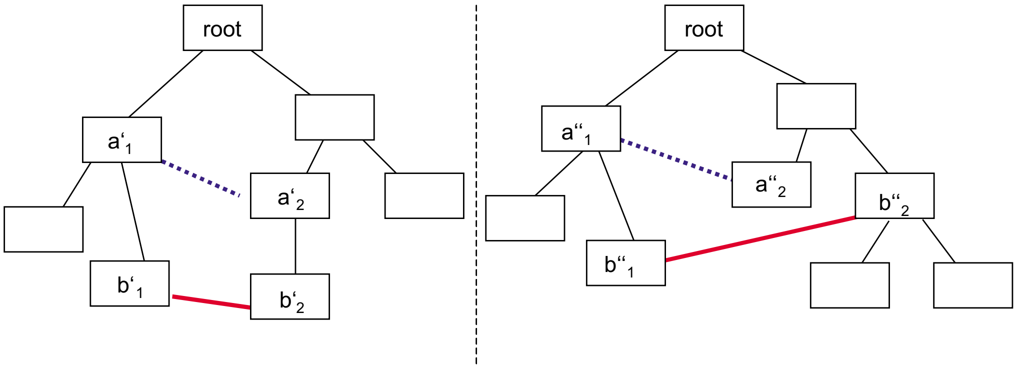


Figure 4: Relation Learning Accuracy

dation is seen as relative to the extent of their agreement such as given by the distance between their least common superconcept, lcs, and the top concept root.[9]

(9) $\text{CLA}(a, b) := \frac{\delta(\text{lcs}(a, b), \text{root})}{\delta(\text{lcs}(a, b), \text{root}) + \delta(a, b)} \in [0, 1].$

The length of the shortest path $\delta(a_s, a_e)$ between a_s and a_e in the taxonomy H is defined via an auxiliary predicate Path that denotes all the valid paths in H.

(10) $\text{Path}(a_0, \ldots, a_n) :\Leftrightarrow \forall i \in 1 \ldots n : (a_{i-1}, a_i) \in H \cup H^{-1}.$

(11) $\delta(a_s, a_e) := \min\{n | a_1, ..., a_{n-1} \in C \wedge \text{Path}(a_s, a_1, ..., a_{n-1}, a_e)\}.$

The only restriction for CLA applies to extremely general relations that use the root concept in their domain or in their range. In our scenario, no such relation appeared in the hand-coded ontology O. Indeed, we found it appropriate to consider such relations as derived from noise that may easily be pruned.

Thus, $\overline{\text{RLA}}$ captures the fact that relations can be introduced at different levels of the taxonomy and that the quality of relations that are learned may vary within a range of degrees.

Example Evaluation. Figure 4 illustrates our definition of the relation learning accuracy with two small examples. On the left hand side of Figure 4 the relation that best matches $d' := (a'_1, a'_2)$ is $r' := (b'_1, b'_2)$. The distances between domain and range concepts count 1 each. The distances $\delta(\text{lcs}(a'_1, b'_1), \text{root})$ and $\delta(\text{lcs}(a'_2, b'_2), \text{root})$ count 1 and 2, respectively. Hence, we compute

(12) $\text{RLA}(d', R) = \text{MA}(d', r') = \sqrt{\frac{1}{1+1} \cdot \frac{2}{2+1}} = \sqrt{\frac{1}{3}} \approx 0.58.$

Similarly, for $d'' := (a''_1, a''_2)$ and $r'' := (b''_1, b''_2)$.

(13) $\text{RLA}(d'', R) = \text{MA}(d'', r'') = \sqrt{\frac{1}{1+1} \cdot \frac{1}{1+2}} = \sqrt{\frac{1}{6}} \approx 0.41.$

Results. An excerpt of our evaluation that surveys the most characteristic results is given in Table 3. We have computed the number of discovered relations D, $\overline{\text{RLA}}$, recall and precision for

[9] Multiple inheritance may result in several least common superconcepts for a pair (a, b). Then we continue using the best value for CLA. All the other definitions remain applicable as they are stated here.

Table 3: Evaluation Results — number of discovered relations, $\overline{\text{RLA}}$, recall, precision

	Confidence			
Support	0.01	0.1	0.2	0.4
0.0001	2429 / 0.55	865 / 0.57	485 / 0.57	238 / 0.51
	66% / 2%	31% / 3%	18% / 3%	2% / 1%
0.0005	1544 / 0.57	651 / 0.59	380 / 0.58	198 / 0.5
	59% / 3%	30% / 4%	17% / 4%	1% / 1%
0.002	889 / 0.6	426 / 0.61	245 / 0.61	131 / 0.52
	47% / 5%	27% / 6%	16% / 6%	1% / 1%
0.01	342 / 0.64	225 / 0.64	143 / 0.64	74 / 0.53
	31% / 8%	19% / 8%	14% / 8%	1% / 1%
0.04	98 / **0.67**	96 / **0.67**	70 / 0.65	32 / 0.51
	13% / 11%	11% / 10%	6% / 7%	0% / 0%
0.06	56 / 0.63	56 / 0.63	48 / 0.62	30 / 0.53
	6% / 9%	6% / 9%	3% / 6%	0% / 0%

varying *support* and *confidence* thresholds. Calculating all relations using a support and confidence threshold of 0 yields 8058 relations, scoring a $\overline{\text{RLA}}$ of 0.51. As expected, both the number of discovered relations D and recall is decreasing with growing support and confidence thresholds. Precision is increasing monotonically at first, but it drops off when so few relations are discovered that almost no one is a direct hit. Higher support thresholds correspond to larger $\overline{\text{RLA}}$ values. Moving confidence thresholds from 0 to 1, $\overline{\text{RLA}}$ peaks between 0.1 and 0.2, but decreases thereafter. This behaviour may be due to our definition of transaction sets and will have to be further explored. The best $\overline{\text{RLA}}$ is reached using a support threshold of 0.04 and a confidence threshold of 0.01 and achieves 0.67 (better than example 12). This constellation also results in the best trade off between recall and precision (13% and 11%). The $\overline{\text{RLA}}$ value of 0.53 remains meaningful, even when recall and precision fall to 0%, due to a lack of exactly matching relations.

Standard deviation ranged between 0.22 and 0.32 in our experiments. Given that our average $\overline{\text{RLA}}$ scored well in the sixties, this means that we had a significant portion of bad guesses, but — what is more important — a large number of very good matches, too. Hence, we may infer that our approach is well-suited for integration into an interactive ontology editor. The reason is that an editor does not require near perfect discovery, but a restriction from a large number of relations, e.g. $283^2 = 80089$ (squared number of concepts leaving out root), to a selection, e.g. a few hundred, that contains a reasonable high percentage of good recommendations.

Random Choice. Finally, we have explored the significance of our $\overline{\text{RLA}}$ measure as compared against a uniform distribution of all possible, viz. 283^2, conceptual relations. The $\overline{\text{RLA}}$ computed from this set was 0.39 and, thus, significantly worse than learning results in our approach. Standard deviation achieved 0.17 and, thus, it was lower than for our discovery approach — the good match by random is indeed very rare. One may note that though the overall mean of 0.39 is still comparatively high (comparable to the one of example (13)), there are relations that score with the minimum, i.e. $\exists d \in D : \text{RLA}(d, R) = 0$, in our ontology.

7 Related Work

As mentioned before, most researchers in the area of discovering conceptual relations have "only" considered the learning of taxonomic relations. To mention but a few, we refer to some fairly recent work, e.g., by Hahn & Schnattinger (Hahn & Schnattinger, 1998) and Morin (Morin, 1999) who used lexico-syntactic patterns with and without background knowledge, respectively, in order to acquire taxonomic knowledge.

Other researchers also pursue a similar principle goal, viz. the semi-automatic engineering of ontologies from text. Our architectural framework (cf. Section 2) provides a comprehensive picture into which these other approaches may be subsumed (Szpakowicz, 1990; Biébow & Szulman, 1999; Faure & Nedellec, 1998).

Regarding the acquisition of non-taxonomic conceptual relations we want to give a somewhat closer look at related approaches. For purposes of natural language processing, several researchers have looked into the acquisition of verb meaning, subcategorizations of verb frames in particular. Resnik (Resnik, 1993) has done some of the earliest work in this category. His model is based on the distribution of predicates and their arguments in order to find selectional constraints and, hence, to reject semantically illegitimate propositions like "The number 2 is blue." His approach combines information-theoretic measures with background knowledge of a hierarchy given by the WordNet taxonomy. He is able to partially account for the appropriate level of relations within the taxonomy by trading off a marginal class probability against a conditional class probability, but he does not give any evaluation measures for his approach. He considers the question of finding appropriate levels of generalization within a taxonomy to be very intriguing and concedes that further research is required on this topic (cf. p. 123f in (Resnik, 1993)) .

Faure and Nedellec (Faure & Nedellec, 1998) have presented an interactive machine learning system called ASIUM, which is able to acquire taxonomic relations and subcategorization frames of verbs based on syntactic input. The ASIUM system hierarchically clusters nouns based on the verbs that they co-occur with and *vice versa*.

Wiemer-Hastings *et al.* (Wiemer-Hastings et al., 1998) aim beyond the learning of selectional constraints, as they report about inferring the meanings of unknown verbs from context. Using WordNet as background knowledge, their system, Camille, generates hypotheses for verb meanings from linguistic and conceptual evidence. A statistical analysis identifies relevant syntactic and semantic cues that characterize the semantic meaning of a verb, e.g. a terrorist actor and a human direct object are both diagnostic for the word "kidnap".

The proposal by Byrd and Ravin (Byrd & Ravin, 1999) comes closest to our own work. They extract named relations when they find particular syntactic patterns, such as an appositive phrase. They derive unnamed relations from concepts that co-occur by calculating the measure for mutual information between terms — rather similar as we do. Eventually, however, it is hard to assess their approach, as their description is rather high-level and lacks concise definitions.

To contrast our approach with the research just cited, we want to mention that all the verb-centered approaches may miss important conceptual relations not mediated by verbs. All of the cited approaches except (Resnik, 1993) neglect the importance of the appropriate level of abstraction. Regarding evaluation, they have only appealed to the intuition of the reader (Byrd & Ravin, 1999; Faure & Nedellec, 1998), focused at a distinguished level in the hierarchy (Wiemer-Hastings et al., 1998) or lacked rigorous measures for evaluation (Resnik, 1993). We have evaluated our approach in blind experiments using two standard and our original $\overline{\text{RLA}}$ measure. The latter has been thoroughly tested for plausibility and validated against the set of all possible relations.

8 Conclusion

We have presented an approach towards learning non-taxonomic conceptual relations from text embedded in a general architecture for semi-automatic acquisition of ontologies. We have evaluated the discovery approach on a set of real world texts against conceptual relations that had been modeled by hand. For this purpose, we used standard measures, viz. precision and recall, but we also developed an evaluation metrics that took into account the scales of adequacy prevalent in our target structures. The evaluation showed that though our approach is too weak for fully automatic discovery of non-taxonomic conceptual relations, it is highly adequate to help the ontology engineer with modeling the ontology through proposing conceptual relations.

For the future much work remains to be done. We want to highlight but two major issues. The naming and the categorization of relations into a relation hierarchy needs to be approached. We want to combine some of the related work on the acquisition of verb meaning with our own proposal in order to approach this objective.

Then, there remains the topic of engineering ontological axioms. Naturally, this is worth several papers on its own. We may just mention that we envision several positions from which to start. We have conceived a principled approach to the engineering of ontological axioms (Staab & Maedche, 2000). Our approach may be extended towards an interactive mode that has been proposed in (Klettke, 1998) for the acquisition of integrity constraints (aka axioms) aiming at the modeling of relational databases. Other than that, we want to explore possibilities offered by inductive logic programming methods — which, of course, presume the availability of corresponding data in order to allow for induction of logical rules.

Acknowledgments. The research presented in this paper has been partially funded by BMBF under grant number 01IN802 (project “GETESS”). We thank our students Raphael Volz and Dirk Wenke who implemented large parts of the learning algorithm and the ontology editor, respectively, and our project partners, in particular Günter Neumann, from DFKI, language technology group, who generously supported us in using their SMES system.

References

Abecker, A., Bernardi, A., & Sintek, M. (1999). Proactive knowledge delivery for enterprise knowledge management. In *SEKE-99: Proceedings of the 11th Conference on Software Engineering and Knowledge Engineering. Kaiserslautern, Germany, June 17-19 1999.*

Biébow, B. & Szulman, S. (1999). TERMINAE: A linguistics-based tool for the building of a domain ontology. In *EKAW ’99 - Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management. Dagstuhl, Germany*, LNCS, pages 49–66, Berlin. Springer.

Byrd, R. & Ravin, Y. (1999). Identifying and extracting relations from text. In *NLDB'99 — 4th International Conference on Applications of Natural Language to Information Systems*.

Faure, D. & Nedellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on adapting lexical and corpus resources to sublanguages and applications*, Granada, Spain.

Hahn, U. & Schnattinger, K. (1998). Towards text knowledge engineering. In *Proc. of AAAI '98*, pages 129–144.

Hajicova, E. (1987). Linguistic meaning as related to syntax and to semantic interpretation. In Nagao, M. (Ed.), *Language and Artificial Intelligence. Proceedings of an International Symposium on Language and Artificial Intelligence*, pages 327–351, Amsterdam. North-Holland.

Hudson, R. (1990). *English Word Grammar*. Basil Blackwell, Oxford.

Klettke, M. (1998). *Acquisition of Integrity Constraints in Databases*. DISDBIS 51. infix, Sankt Augustin, Germany. In German.

Maedche, A., Schnurr, H.-P., Staab, S., & Studer, R. (2000). Representation language-neutral modeling of ontologies. In Frank (Ed.), *Proceedings of the German Workshop "Modellierung-2000". Koblenz, Germany, April, 5-7, 2000*. Fölbach-Verlag.

Morin, E. (1999). Automatic acquisition of semantic relations between terms from technical corpora. In *Proc. of the Fifth International Congress on Terminology and Knowledge Engineering - TKE'99*.

Neumann, G., Backofen, R., Baur, J., Becker, M., & Braun, C. (1997). An information extraction core system for real world german text processing. In *ANLP'97 — Proceedings of the Conference on Applied Natural Language Processing*, pages 208–215, Washington, USA.

Resnik, P. (1993). *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, University of Pennsylania.

Romacker, M., Markert, M., & Hahn, U. (1999). Lean semantic interpretation. In *Proc. of IJCAI-99*, pages 868–875.

Schnurr, H.-P. & Staab, S. (2000). A proactive inferencing agent for desk support. In *Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes*, Stanford, CA, USA. AAAI Technical Report, Menlo Park.

Srikant, R. & Agrawal, R. (1995). Mining generalized association rules. In *Proc. of VLDB '95*, pages 407–419.

Staab, S. & Maedche, M. (2000). Axioms are Objects, too - Ontology Engineering beyond the modeling of Concepts and Relations. Technical Report 400, Institute AIFB, Karlsruhe University.

Szpakowicz, S. (1990). Semi-automatic acquisition of conceptual structure from technical texts. *International Journal of Man-Machine Studies*, 33.

Wiederhold, G. (1993). Intelligent integration of information. In *SIGMOD-93*, pages 434–437.

Wiemer-Hastings, P., Graesser, A., & Wiemer-Hastings, K. (1998). Inferring the meaning of verbs from context. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*.