Semi-Automatic Engineering of Ontologies from Text

A. Maedche and S. Staab

Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
E-mail: {maedche,staab}@aifb.uni-karlsruhe.de

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 present a general architecture for discovering conceptual structures and engineering ontologies. Based on the architecture 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.

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 [19, 1, 17]. 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 propos-als have been made to facilitate ontology engineering through automatic discovery from domain data, domain-specific natural language texts in particular (cf. [3, 4, 5, 11, 13, 20]). However, we see two pitfalls occur in most of these seminal approaches. First, these investigations 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 [19, 1, 17], 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-

---

[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.

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

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. We conclude with a survey of related work and a short remark on the acquisition of ontological axioms, A.

2 System Architecture

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. [9]). 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.
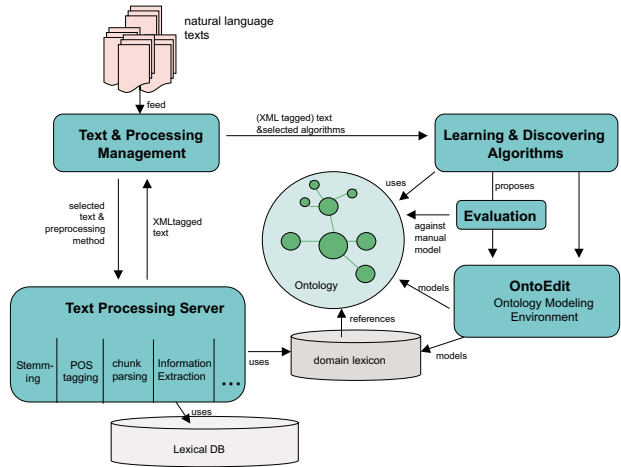

Figure 1. Architecture of the Ontology Learning Environment

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

---

[3] For ease of presentation we mostly give English examples, however, our evaluation is based on our implementation that processes German texts.
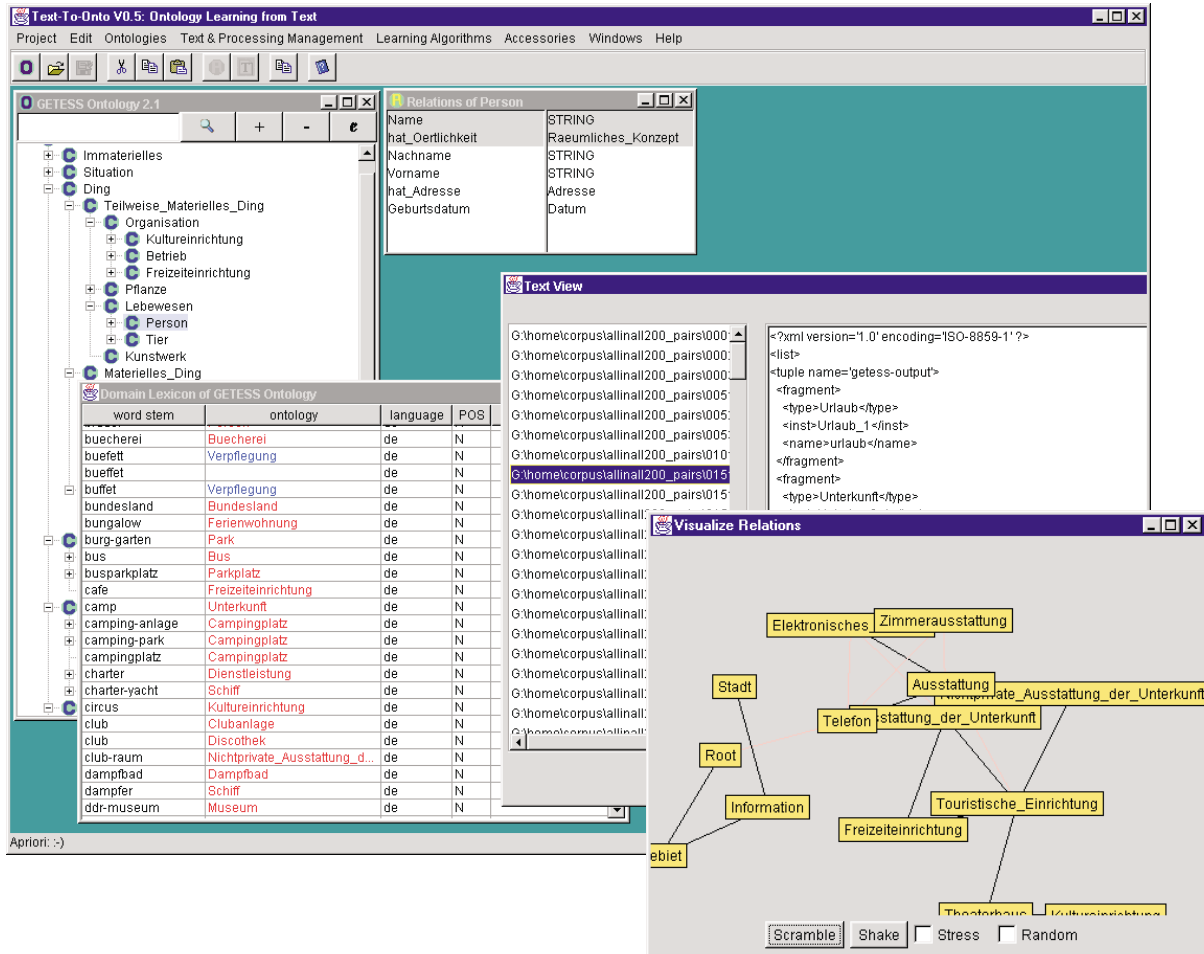
Figure 2. The Text-To-Onto Ontology Learning Environment

chunk parsing. The domain-specific part of the lexicon (abbreviated "domain lexicon"; cf. left lower part of Figure 2) associates word 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 of the applied algorithms such as described in [10] is performed in a submodule based on the results of the learning algorithm.

Ontology Engineering Environment. The Ontology Engineering 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 the ontology, the engineering environment provides some additional features for the purpose of documentation, maintenance, and ontology exchange. OntoEdit internally stores ontologies using an XML serialization of the ontology model.

---

[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 [16].

3   Shallow Text Processing

Our approach has been implemented on top of SMES (Saarbrücken Message Extraction System), a shallow text processor for German (cf. [12]) 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 to provide the reader with a comprehensive picture of what underlies our system.

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 subcategorization frames in the lexicon.
Dependency Relations. Our primary output derived from SMES consists of *dependency relations* [7] found through lexical analysis (compound processing) and through parsing 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. [6]). 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. [14] 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

---

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

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 [15]. 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.*

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

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

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 {Hotel}⇒{Address} is equal or, much more probably, (far) less than 1%,

while rules about rooms and their furnishing or their style, like {Room}⇒{Bed}, might achieve ratings of several percentage points. This means that an important relationship between {Hotel} and {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 {Hotel}⇒{Address} and {Room}⇒{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).

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

Table 1. 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

concepts, $C := \{a_i\}$, and the domain-specific part of the lexicon has contained 486 entries referring to one of these concepts.
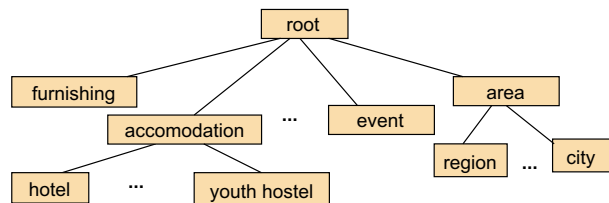

Figure 3. An example scenario

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 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 [5] and Morin [11] 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 [18, 2, 4]. For example in [2] the system TERMINAE for building a domain ontology using a terminology- based approach is described. The underlying techniques are restricted to statistical term occurrences, which are also a part of our system Text-To-Onto. More advanced machine learning techniques are applied in the ASIUM system presented by Faure and Nedellec [4]. The system 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*. However, this approach and the algorithms developed may easily be integrated into our framework, so that the acquired ontology may be refined further.

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 [13] 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 [13]) .

Wiemer-Hastings *et al.* [20] 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 [3] 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 [13] neglect the importance of the appropriate level of abstraction. Regarding evaluation, they have only appealed to the intuition of the reader [3, 4], focused at a distinguished level in the hierarchy [20] or lacked rigorous measures for evaluation [13]. We have evaluated our approach in blind experiments using two standard and our original $\overline{RLA}$ measure (cf. [10] for a more detailed description). The latter has been thoroughly tested for plausibility and validated against the set of all possible relations.

7 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 [16]. Our approach may be extended to-

wards an interactive mode that has been proposed in [8] 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.

References

[1] A. Abecker, A. Bernardi, and M. Sintek. 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*, 1999.

[2] B. Biébow and S. Szulman. 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, 1999. Springer.

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

[4] D. Faure and C. Nedellec. 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, 1998.

[5] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proc. of AAAI '98*, pages 129–144, 1998.

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

[7] R. Hudson. *English Word Grammar*. Basil Blackwell, Oxford, 1990.

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

[9] A. Maedche, H.-P. Schnurr, S. Staab, and R. Studer. Representation language-neutral modeling of ontologies. In U. Frank, editor, *Proceedings of the German Workshop "Modellierung-2000". Koblenz, Germany, April, 5-7, 2000*. Fölbach-Verlag, 2000.

[10] A. Maedche and S. Staab. Discovering Conceptual Relations from Text. Technical Report 399, Institute AIFB, Karlsruhe University, 2000.

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

[12] G. Neumann, R. Backofen, J. Baur, M. Becker, and C. Braun. 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, 1997.

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

[14] M. Romacker, M. Markert, and U. Hahn. Lean semantic interpretation. In *Proc. of IJCAI-99*, pages 868–875, 1999.

[15] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. of VLDB '95*, pages 407–419, 1995.

[16] S. Staab and A. Maedche. Axioms are Objects, too - Ontology Engineering beyond the modeling of Concepts and Relations. Technical Report 400, Institute AIFB, Karlsruhe University, 2000.

[17] S. Staab and H.-P. Schnurr. Smart Task Support through Proactive Access to Organizational Memory. *Journal of Knowledge-based Systems*, to appear, 2000.

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

[19] G. Wiederhold. Intelligent integration of information. In *SIGMOD-93*, pages 434–437, 1993.

[20] P. Wiemer-Hastings, A. Graesser, and K. Wiemer-Hastings. Inferring the meaning of verbs from context. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, 1998.