Mining Ontologies from Text

Alexander Maedche and Steffen Staab

AIFB, Univ. Karlsruhe, D-76128 Karlsruhe, Germany
{maedche, staab}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS

Abstract Ontologies have become an important means for structuring knowledge and building knowledge-intensive systems. 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 our generic architecture we describe a case study for mining ontologies from text using methods based on dictionaries and natural language text. The case study has been carried out in the telecommunications domain. Supporting the overall text ontology engineering process, our comprehensive approach combines dictionary parsing mechanisms for acquiring a domain-specific concept taxonomy with a discovery mechanism for the acquisition of non-taxonomic conceptual relations.

1 Introduction

Ontologies[1] have shown their usefulness in application areas such as intelligent information integration [23], information brokering [20] and natural-language processing [21], to name but a few. However, their wide-spread usage is still hindered by ontology engineering being rather time-consuming and, hence, expensive.

A number of proposals have been made to facilitate ontological engineering through automatic discovery from domain data, domain-specific natural language texts in particular (cf. [1,3,5,13,14,16,24]). However, most approaches have "only" tackled one step in the overall ontology engineering process, e.g. the acquisition of concepts, the establishment of a concept taxonomy or the discovering of non-taxonomic conceptual relationships, whereas one must consider the overall process when building real-world applications.

In this paper we describe a case study for mining ontologies from textual resources, *viz.* from technical dictionaries and from domain texts, where we

[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.

consider all three before-mentioned steps. For this purpose we combine existing techniques for the acquisition of concepts and a concept taxonomy with a new approach for mining non-taxonomic conceptual relationships from natural language in an integrated framework for manual and semi-automatic ontology engineering.

The remainder of the paper is as follows. In Section 2 we will give an overview of the overall system architecture, in particular about which linguistic processing has been done and how discovered conceptual structures are added to the ontology using a graphical ontology engineering environment. Subsequently, we will focus on the techniques for acquiring concepts and concept hierarchies which are an essential part for the algorithm discovering non-taxonomic conceptual relations. This algorithm will be presented in Section 4. An example will show some promising results we obtained applying our mechanisms for mining ontologies from text. Before we conclude we give an overview of related work in Section 5.

2 Architecture

The purpose of this section is to give an overview of the architecture of our approach. 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. [12]). 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.

2.1 Text & Processing Management Component

The ontology engineer uses the Text & Processing Management component to select domain resources (dictionaries, 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 described in subsection 2.3.

2.2 Text Processing Server

The Text Processing Server comprises 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; cf. [15]). SMES is a system that performs syntactic analysis on natural language documents. In general, the Text

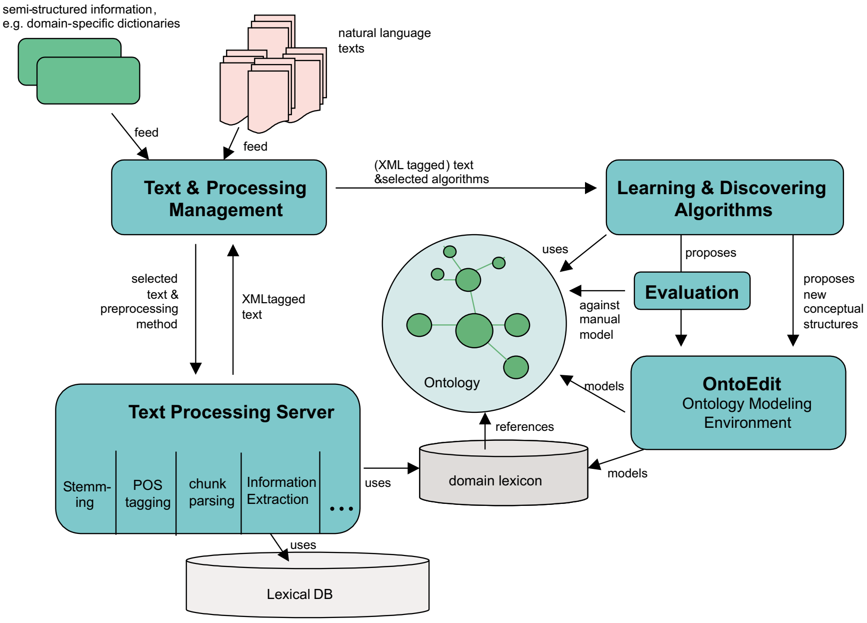

semi-structured information, e.g. domain-specific dictionaries
natural language texts
feed
feed
Text & Processing Management
(XML tagged) text &selected algorithms
Learning & Discovering Algorithms
uses
proposes
Evaluation
against manual model
proposes new conceptual structures
selected text & preprocessing method
XMLtagged text
Ontology
models
OntoEdit
Ontology Modeling Environment
references
Text Processing Server
Stemm-ing
POS tagging
chunk parsing
Information Extraction
...
uses
domain lexicon
models
uses
Lexical DB

Figure 1. Architecture of the Ontology Learning Environment

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.

SMES 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 "normalized" 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 delivering a high recall on the number of dependency relations occuring in a text, is also guaranteed.

The Architecture of SMES comprises a *tokenizer* based on regular expressions, a *lexical analysis* component including a word and a domain *lexicon*, 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 "Baden-Wuerttemberg"[2], and to expand abbreviations.

Lexicon. The lexicon contains more than 120.000 stem entries and more than 12,000 subcategorization frames describing information used for lexical analysis

[2] Baden-Wuerttemberg is a region in the south west of Germany.

and chunk parsing. Furthermore, the domain-specific part of the lexicon associates word stems with concepts that are available in the concept taxonomy.

The reader may note that at the beginning there are no or only few mappings from word stems to (some few, domain-independent) concepts available in the domain lexicon. Only with the extension of the ontology the domain-specific part of the lexicon is augmented, too.[3] At the beginning the ontology engineer uses simple means, e.g. word counts, in order to establish new concepts and their linkages to word stems. By doing so, she leverages the linguistic processing and, thus, the further knowledge discovery process in subsequent stages.

Lexical Analysis uses the lexicon 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 "Deutsche Telekom AG" 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 "Deutsche Telekom AG" is associated with the concept TKCompany.
4. Finally, part-of-speech tagging disambiguates the reading returned from morphological analysis of words or complex expressions using the local context.

Lexical analysis is the first of two primary outputs from SMES that we exploit. It returns "normalized" readings for different word forms (e.g., singular vs. plural) that we want to abstract from in order to add a corresponding concept to the ontology.

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.

The chunk parser of SMES returnes the second primary output that we use, *viz. dependency relations* [9] found through lexical analysis (compound processing) and through parsing at the phrase and sentential level. We take advantage of the fact that syntactic dependency relations coincide rather closely with semantic relations holding between the very same entities (cf. [6]). Thus, we consider

[3] In the future, we also want to extend the lexicon proper during domain adaptation.

syntactic results as the signposts that points our discovery algorithms into the direction of semantic relationships. We feed 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, once we have the linkages between "France Telecom" and "Paris" denoting instances of Company and City, respectively, in example (1), we may conjecture a semantic relationship between Company and City. The motivation is derived from the dependential relationships between "France Telecom", "in", and "Paris". The preposition "in" acts as a mediator that incurs the conceptual pairing of Company with City (cf. [17] for a comprehensive survey of mediated conceptual relationships).

(1) *France Telecom* in *Paris* offers the new DSL technology.

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.

Thus, these heuristics complement the output produced by the chunk parser. **To sum up**, linguistic processing outputs "normalized" terms and sets of concept pairs, $CP := \{(a_{i,1}, a_{i,2}) | a_{i,j} \in C\}$. Normalization is based on lexical analysis and the coupling of concepts is motivated through various direct and mediated linguistic constraints or by several general or domain-specific heuristic strategies.

2.3 Learning & Discovering component

The Learning & Discovering component uses various algorithms on the annotated texts:

1. Conventional term extraction mechanisms are applied to extract relevant terms from the corpus.

2. An approach for mining a concept taxonomy from a dictionary, which is based on regular expression-based pattern matching algorithms, described in further detail in Section 3
3. An approach for mining non-taxonomic relations, that uses the learning algorithm for discovering generalized association rules described in Section 4.

Conceptual structures that exist at learning time (e.g. concepts or a concept taxonomy) may be incorporated into the learning algorithms as background knowledge. The evaluation of the applied algorithms such as described in [13] is performed in a submodule based on the results of the learning algorithm.

2.4 Ontology Engineering Environment OntoEdit

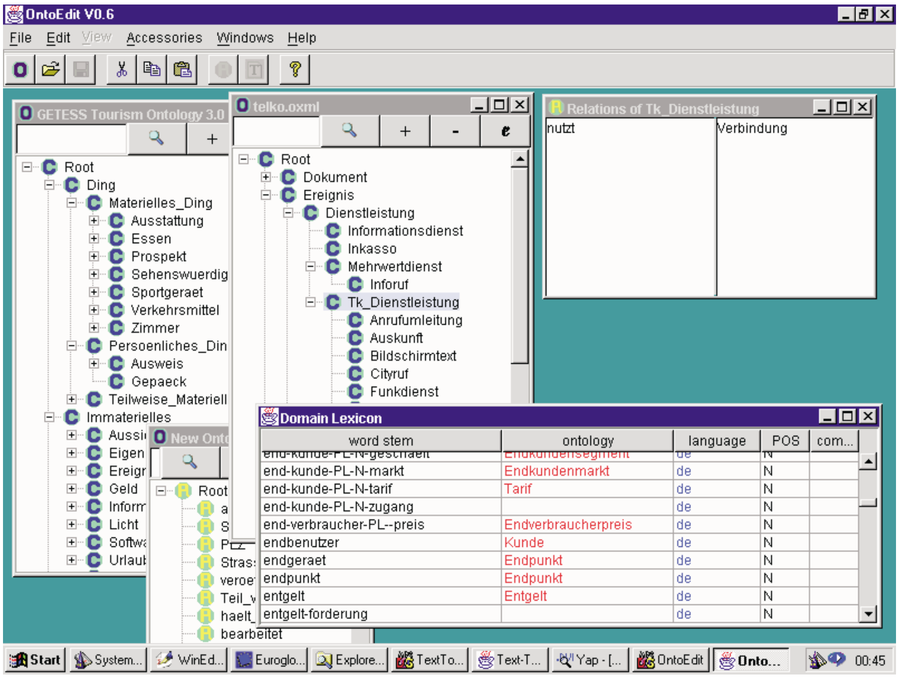
The Ontology Engineering Environment OntoEdit, a submodule of the Ontology Learning Environment "Text-To-Onto" (cf. Figure 2), supports the ontology engineer in semi-automatically adding newly discovered conceptual structures to the ontology.[4] In addition to core capabilities for structuring the ontology, 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. OntoEdit accesses an inference engine that is based on Frame-Logic.[5]

2.5 System Wrap-up

The principle idea of our framework is based on applications of knowledge discovery techniques based on input from linguistic processing in a semi-automatic bootstrapping approach. The learning mechanisms in our system do not determine the complete structure, but they are only meant to help the ontology engineer with building a domain ontology by giving recommendations for adding concepts or relations. The system is also not intended to be used in a pipeline fashion, but rather we conceive that simple methods should be exploited first in order to determine the scope of the ontology and the set of relevant concepts. With the extension of the ontology, conceptual *and* linguistic resources are augmented and, thus, they nourish more complex and fruitful linguistic processing and knowledge discovery in subsequent passes through the ontology learning and engineering cycle.

[4] A comprehensive description of the ontology engineering system OntoEdit and the underlying methodology is given in [22].

[5] F-Logic is a frame-logic representation language conceived by [10]. In the implementation by Angele and Decker that we use, F-Logic is a proper subset of first-order predicate logic. Concepts and relations are reified and, hence, may be treated as first-order objects over which quantification is possible. For efficient processing, F-Logic is translated into a datalog-style representation (*cf.* [11,2]).



word stem	ontology	language	POS	com...
end-kunde-PL-N-geschaeft	Endkundensegment	de	N	
end-kunde-PL-N-markt	Endkundenmarkt	de	N	
end-kunde-PL-N-tarif	Tarif	de	N	
end-kunde-PL-N-zugang		de	N	
end-verbraucher-PL--preis	Endverbraucherpreis	de	N	
endbenutzer	Kunde	de	N	
endgeraet	Endpunkt	de	N	
endpunkt	Endpunkt	de	N	
entgelt	Entgelt	de	N	
entgelt-forderung		de	N	

Figure 2. OntoEdit

3 Mining a concept taxonomy from a telecommunications dictionary

In order to provide a starting set of concepts and their taxonomic relations for the domain ontology of our case study, we have exploited the structuring of a freely available dictionary from the telecommunications domain in the first step. In order to make use of the resulting ontology as input for the discovery of further conceptual relations (cf. Section 4), we also had to acquire the mapping between concepts and words.

Example

A dictionary containing natural-language definitions of terms in the telecommunications domain, which is freely available at *http://www.interest.de/online/tkglossar/index.html*, served as a good starting point for the case study. The given 1465 HTML pages were downloaded and transformed into our predefined XML representation for dictionaries, such as given in the following small example:

```
<termEntry>
   <admin>
```

```
     <entrynumber>1328</entrynumber>
  </admin>
  <term lang='Deutsch'> Kommunikationsserver
        <description type='Definition'>
          <descriptionText>
             Zentrale Funktionseinheit, welche fuer
             mehrere Benutzer Kommunikationsdienste erbringt.
          </descriptionText>
        </description>
    </term>
</termEntry>
```

Every entry has been defined as a concept and a corresponding domain lexicon entry of this concept (reduced to its word stem) has been generated using the Text Processing Server lexical analysis. The definitions of the terms have also been processed using the Text Processing Server.

Similar to the work described in [8,14], we have defined several lexico-syntactic patterns in the form of regular expressions for extracting ISA relations between concepts on the given processed and normalized dictionary definitions. In our small example above the following simple pattern, which is expressed in natural language here for ease of presentation, matched:

"the last NP of the definition before the last comma represents a hypernym of the concept to be defined"

The patterns have resulted in ISA relations, such as between the concept Kommunikationsserver (engl. communication server) and Funktionseinheit (engl. functional unit).

However, we have to emphasize that for building a representative domain ontology, the described dictionary parsing mechanisms are not sufficient. Typically, domain-specific dictionaries describe terms only at a very detailed technical level focusing on the leaf concepts of the taxonomy. For instance, the above mentioned dictionary lacked many important concepts, such as private customer and business customer. For this reason, we have also applied term extraction mechanisms based on the tfidf measure [18] on the given corpus in order to propose frequent terms as candidate concepts. These concepts were then added manually to the domain ontology.

This mixed approach using a combination of automatic extraction mechanisms and user modeling resulted in an core ontology with 265 concepts connected through 312 ISA relations. Additionally, 620 domain lexicon entries mapping words to concepts have been brought into the system.

4 Mining Generic Relations from Text

Our text mining mechanism for discovering relations between concepts is based on the algorithm for discovering generalized association rules proposed by Srikant and Agrawal [19]. Their algorithm is used for well-known applications of data mining, viz. finding associations that occur between items, e.g. supermarket products, in a set of transactions, e.g. customers' purchases. The algorithm aims at descriptions 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.*

$$(2)\ \ \text{support}(X_k \Rightarrow Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{n}$$

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

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.

The reader may note that 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, let us assume a set of 100 texts each describing a particular client in detail. Each private client might come with an address, but it might also have an elaborate description of the different types of private telecomunication services and different calling types 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 {PrivateClient}⇒{Address} is equal or, much more probably, (far) less than 1%, while rules about telecommunication services and different calling types might achieve ratings of several percentage points. This means that an important relationship between {PrivateClient} 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 {PrivateClient}⇒{Address} and {Service}⇒{CallingType} 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.

Example

For the purpose of illustration, we here give a comprehensive example, which is based on our actual experiments. We have generated a text corpus by crawling texts from several WWW providers for telecommunications information (URL: http://www.TK-news.de/). The corpus describes actual objects, like telecommunication companies, new technologies, telecommunication services, and trends, such as given in the following example sentences.

(4) a. France Telecom bietet als erster Telekommunikationsdienstleister das *DSL-Netz* mit maximaler *Uebertragungsgeschwindigkeit*.
b. Die *Swiss Telekom Beteiligungsgesellschaften* erschweren den Fortschritt.
c. Laut interner Information wird *France Telecom* mit der *BCDM AG* mergen.
d. Alle *Basisanschluesse* sind mit *Kabel*, *Telefon* und *PC-Karte* ausgestattet.

Processing the example sentences (4a) and (4b), SMES (Section 2) outputs dependency relations between the terms, which are indicated in *slanted fonts* (and some more). In sentences (4c) and (4d) 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$_2$	$a_{i,2}$
DSL-Netz	DSLNetz	*Ueb.geschwindigkeit*	Uebgeschwindigkeit
Swiss Telekom	TKCompany	*Bet.gesellschaft*	Bet.gesellschaft
France Telecom	TKCompany	*BCDM AG*	TKCompany
Basisanschluss	Basisanschluss	*Kabel*	Kabel

The algorithm for learning generalized association rules (cf. Section 4) uses our semi-automatically generated 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, it discovered a large number of interesting and important non-taxonomic conceptual relations.

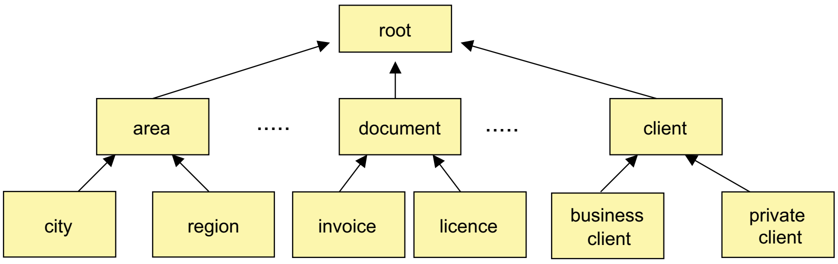


Figure 3. An example scenario

A few of them are listed in Table 2. Note that in this table we also list a conceptual pair, viz. (private client, city), that is not presented to the user, but which is pruned. The reason is that there is an ancestral association rule, viz. (client, city), with higher confidence and support measures.

Table 2. Examples of discovered relations

Discovered relation	Confidence	Support
(market, tariff)	0.38	0.04
(connection, price)	0.39	0.03
(TKCompany, TKCompany)	0.5	0.1
(client, city)	0.39	0.03
~~(private client, city)~~	~~0.29~~	~~0.02~~

5 Related Work

The objective of our framework is to facilitate ontology engineering from texts in real-world settings through several information extraction and learning approaches. Thus, we had to face (i) the discovery of relevant concepts, (ii) their organization in a taxonomy, and (iii) the non-taxonomic relationsships between concepts.

In our actual case study, we have employed a three-step approach. We have exploited some fairly well-known methods for concept discovery and organization, such as standard statistics-based approaches for term extraction [18] and the use of lexico-syntactic patterns on machine-readable dictionaries [8,14].

Based on the concept hierarchy from the first two steps, we have set a new method for the discovery of non-taxonomic relations on top. Regarding this part of our work, we want to give a more detailed survey of related work.

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 [14] who used lexico-syntactic patterns with and without background knowledge, respectively, in order to acquire taxonomic knowledge.

For purposes of natural language processing, several researchers have looked into the acquisition of verb meaning, subcategorizations of verb frames in particular. Resnik [16] 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. 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 [16]) .

Faure and Nedellec [3] 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.* [24] 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 [1] 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 [16] neglect the importance of the appropriate level of abstraction.

6 Conclusion

In this paper we have presented an approach towards mining ontologies from natural language. We have considered a domain-specific dictionary as well documents taken from the telecommunications domain as relevant resources for the difficult task of ontology learning.

For the future much work remains to be done. First, we need to investigate what specific types of linguistic and heuristic output are best suited to optimize performance. Maybe chunk parsing does not even help so much, but noun phrase recognition does, or *vice versa*. Second, we are planning a study to investigate our defined evaluation and similarity measures precision, recall, and $\overline{\text{RLA}}$ described in [13] for that human modelers achieve when they are given the same task as our discovery mechanism. Third, we will have to investigate the influence of different transaction definitions (cf. Section 4). Fourth, several existing ontologies such as WordNet [4] and the german counterpart GermaNet [7] have to be integrated as a core resource into the cyclic approach and mechanisms for pruning ontologies to the relevant domain have to be developed. Finally, and probably the most intricate, we want to approach not only the learning of the existence of relations, but also their names and types.

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

References

1. 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.
2. S. Decker. On domain-specific declarative knowledge representation and database languages. In A. Borgida, V. Chaudri, and M. Staudt, editors, *KRDB-98 — Proceedings of the 5th Workshop Knowledge Representation meets DataBases, Seattle, WA, 31-May-1998*, 1998.
3. 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.
4. Christiane Fellbaum. *WordNet – An electronic lexical database.* MIT Press, Cambridge, Massachusetts and London, England, 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. B. Hamp and H. Feldweg. Germanet - a lexical-semantic net for german. In *Proceedings of* ACL *workshop Automatic Information Extraction and Building of Lexical Semantic Resources for* NLP *Applications, Madrid.*, 1997.
8. M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France*, 1992.
9. R. Hudson. *English Word Grammar.* Basil Blackwell, Oxford, 1990.
10. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42, 1995.
11. J. W. Lloyd and R. W. Topor. Making Prolog more expressive. *Journal of Logic Programming*, 1(3), 1984.
12. A. Maedche, H.-P. Schnurr, S. Staab, and R. Studer. Representation language-neutral modeling of ontologies. In Frank, editor, *Proceedings of the German Workshop "Modellierung-2000". Koblenz, Germany, April, 5-7, 2000.* Fölbach-Verlag, 2000.
13. A. Maedche and S. Staab. Discovering conceptual relations from text. In *Proceedings of ECAI-2000.* IOS Press, Amsterdam, 2000.
14. 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.
15. 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.
16. P. Resnik. *Selection and Information: A Class-based Approach to Lexical Relationships.* PhD thesis, University of Pennsylania, 1993.
17. M. Romacker, M. Markert, and U. Hahn. Lean semantic interpretation. In *Proc. of IJCAI-99*, pages 868–875, 1999.
18. K. Sparck-Jones and P. Willett, editors. *Readings in Information Retrieval.* Morgan Kaufmann, 1997.
19. R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. of VLDB '95*, pages 407–419, 1995.
20. S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. In *WWW9 - Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000.* Elsevier, 2000.
21. S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, and B. Wrenger. Getess - searching the web exploiting german texts. In *CIA '99 - Proceedings of the 3rd international Workshop on Cooperating Information Agents. Upsala, Sweden, July 31-August 2, 1999*, LNAI 1652, pages 113–124. Springer, 1999.
22. S. Staab and A. Maedche. Ontology engineering beyond the modeling of concepts and relations. In *Proceedings of the ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods*, 2000.
23. G. Wiederhold and M. Genesereth. The conceptual basis for mediation services. *IEEE Expert / Intelligent Systems*, 12(5):38–47, September/October 1997.
24. 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.