A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet

Joerg-Uwe Kietz, Alexander Maedche, Raphael Volz

Swisslife Information Systems Research Lab, Zuerich, Switzerland
uwe.kietz@swisslife.ch
http://www.swisslife.ch
AIFB, Univ. Karlsruhe, D-76128 Karlsruhe, Germany
{maedche, volz}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS

Abstract The focused access to knowledge resources like intranet documents plays a vital role in knowledge management and supports in general the shifting towards a *Semantic Web*. Ontologies act as a conceptual backbone for semantic document access by providing a common understanding and conceptualization of a domain. Building domain-specific ontologies is a time-consuming and expensive manual construction task. This paper describes our actual and ongoing work in supporting semi-automatic ontology acquisition from a corporate intranet of an insurance company. We present a comprehensive architecture and generic method for discovering a domain-tailored ontology from given intranet resources.

1 Introduction

The amount of information available to corporate employees has grown drastically with the use of intranets. Unfortunately this growth of available information has made the *access* to useful or necessary information much more difficult due to the fact that the access is usually based on keyword searching or even browsing. Keyword searching results in a lot of irrelevant information as a term can have different meanings in distinct contents, e.g. “Zürich” refers to the name of a town as well as the name of an insurance company. Presently it is quite difficult to provide this information to the search engine, e.g. exclude all the information about the town “Zürich” without losing information about the insurance company “Zürich”. Also the query provided by the user does not always carry the intented meaning. For example, a user looking for “beauty care” will not find any information about “hair care”, as the system must know about the fact, that “hair care” is a specialization of “beauty care”. The focused access to knowledge resources like intranet documents plays a vital role in knowledge management and supports in general the shifting towards a *Semantic Web*, the Next-Generation Web. The project ON-TO-KNOWLEDGE [7] builds an ontology-based tool environment to perform knowledge management dealing with large numbers of heterogeneous, distributed and semi-structured documents as found within large intranets and the World-Wide Web. In this project ontologies play a key role by providing a common understanding of the domain. Semantically annotated documents are accessed using the vocabulary provided by a domain-specific ontology. Providing the user with an access method based on ontological terms instead of keywords has several advantages. First, the abstraction given by the ontology provides that the user does not have to deal

with document-specific representations. Second, by this abstraction robustness towards changes in content and format of the accessed documents is gained.

Currently, the required domain-specific ontologies for ON-TO-KNOWLEDGE are built manually using graphical means available in ontology engineering tools like *OntoEdit* [26] or *Protégé* [8]. Using such tools simplifies ontology construction and maintenance. However, the wide-spread usage of ontologies is still hindered by the time-consuming and expensive manual construction task. Within ON-TO-KNOWLEDGE our work evaluates semi-automatic ontology construction from intranet resources as an alternative approach to manual ontology engineering. Based on the assumption that most concepts and conceptual structures of the domain as well the companies terminology are described in documents, applying knowledge acquisition from text for ontology design seems to be promising. In the recent years 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. [3,6,10,20,23,30]). However, it lacks an overall framework of ontology acquisition from text in which these approaches could be embedded. Our work gives a generic architecture, an acquisition methodology and several new approaches for acquiring concepts and relations from intranet resources. Eventually the extraction of ontologies from text using our approach yields to additional benefits for ON-TO-KNOWLEDGE because the information required for the semantic annotation of documents can be provided as a side effect of the extraction process. This task requires that the domain-specific ontology must be able to adopt to any changes in content. Our approach is cyclic to be able to cope with any changes in content.

Our approach is based on different heterogeneous intranet sources: First, a generic core ontology is used as a top level structure for the domain-specific goal ontology. Second, domain-specific concepts are acquired and classified into the concept taxonomy from a dictionary that contains important corporate terms described in natural language. Third, we use a domain-specific and a general corpus of texts to remove concepts that were domain-unspecific. The removal of concepts follows the heuristic that domain-specific concepts must be more frequent in a domain-specific corpus than in generic texts.

Additionally, we learned non-taxonomic relations between concepts by analyzing the aforementioned intranet documents. We used a multi-strategy approach in learning to level the specific advantages and drawbacks of different learning methods. Several methods were applied with the possibility to combine their results.

The paper is organized as follows. Section 2 describes the overall architecture of the system and explains our notion of ontologies. Section 3 discusses the methodology applied to acquire a domain-specific ontology. Section 4 highlights the applied learning mechanisms. Section 5 demonstrates some preliminary results. Before we conclude, Section 6 points out further directions for our work and acknowledges other contributors to the work.

2 Architecture

The work described in this paper extends the general architecture for semi-automatic ontology engineering from natural language that has been described previously [17]. The architecture comprises of components for resource processing & management, natural language processing, an algorithm library for ontology learning and an ontology

repository as well as tools for manual ontology engineering and inferencing. Along this lines, our system follows the *balanced cooperative modeling* paradigm established by Morik [19]. Her work describes the interaction between knowledge acquisition and machine learning, where each modeling step can be done either by human or by machine. Also, existing knowledge is incorporated into learning algorithms and the output of an algorithm is proposed to the user.

Our notion of ontologies is closely associated to the notion described within the ontology interchange and inference layer OIL [13]. From the expressive power it is equivalent to the CLASSIC-$\mathcal{ALN}$ [2] description logic. OIL combines three important aspects provided by three different communities, namely Description Logics (providing formal semantics and efficient reasoning support), Frame-Based Systems and web standards. XML is used as a serial syntax definition language, describing knowledge in terms of concepts and role restrictions (i.e. all- and cardinality-restrictions as in the DL $\mathcal{ALN}$). Also, relations are regarded as an independent entity whose domain and range concepts can be restricted.

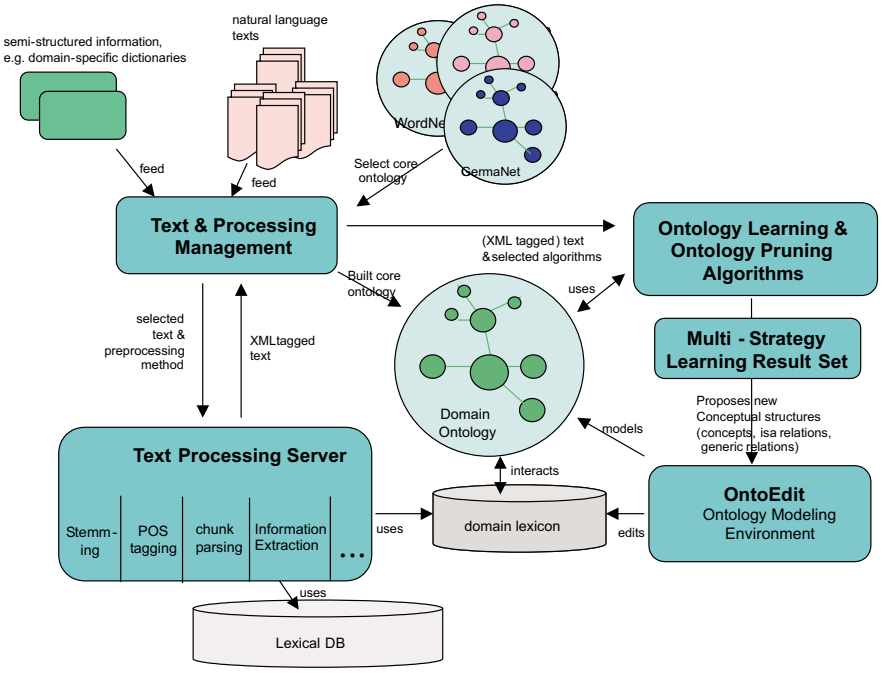

Figure1. Architecture of the Ontology Learning Approach

2.1 Resource management & processing component

As our approach uses different resources several mechanisms for integrating them into our system must to be available to the ontology engineer. This component supports efficient handling and processing of input sources, namely semi-structured information contained in dictionaries, natural language documents and existing ontologies (cf. Section 4.1).

2.2 Natural language processing

The necessary natural language processing functionality is provided by the system SMES (Saarbrücken Message Extraction System), a shallow text processor for German (cf. [22,21]). We will give a short survey on SMES in order to provide the reader with a comprehensive picture of what underlies our system. This is a generic natural language processing component that adheres to several principles that are crucial for our objectives.

The architecture of SMES comprises of a *tokenizer* based on regular expressions, a *lexical analysis* component including a *word and a domain lexicon*, and a *chunk parser*. The tokenizer scans the text in order to identify boundaries of words, complex expressions like "$20.00" and to expand abbreviations. The user is able to provide domain-specific compounds like department names (e.g.: "CC/ITRD"[1]. The lexicon contains more than 700,000 stem entries and more than 12,000 subcategorization frames describing information used for lexical analysis and chunk parsing. Furthermore, the domain-specific part of the lexicon associates word stems with concepts that are available in the concept taxonomy. *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], *(2)*, recognition of named entities, *(3)*, retrieval of domain-specific information, and, *(4)*, part-of-speech tagging. While the steps (1),(2) and (4) can be a viewed as standard for information extraction approaches (cf. [1,21]), the step (3) is of specific interest for our task. This 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 "Federal Law Regulating Social Benefits" can be associated with the concept Federal Law. SMES includes a *chunk parser* based on weighted finite state transducers to efficiently process phrasal and sentential patterns. The parser works on the phrasal level, before it analyzes the overall sentence. The results of each step are annotated in XML-tagged text and can be used independently.

2.3 Algorithm library for ontology learning

Ontology engineers have to perform several steps to create ontologies. Independent of the design principles one adheres (e.g. in [9,27]) several steps can be identified. Each step of ontology engineering is supported by several learning algorithms contained in the library of our system. They operate on the extracted information and are used for two tasks: One task is the acquisition of new structures, the second task is the evaluation of given structures.

As mentioned before one of the core capabilities of our system is *multi-strategy learning* (as described in [18]). All learning methods use a common result structure. Therefore the engineer can combine results and is supported in balancing between advantages and disadvantages of different learning methods. Due to this combination and balancing the complex task of ontology engineering is fitted better.

[1] *CC/ITRD* is the acronym for the Swiss Life Information Systems Lab used within the company.

[2] 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.

2.4 Ontology Engineering

We argue, that learning helps a lot but is not sufficent (yet). Thus, mechanisms for ergonomic, manual ontology modeling must still be provided by any semi-automatic ontology engineering system. We used OntoEdit[3] within our system. It allows editing and browsing of existing as well as discovered ontological structures and has the possibility of defining axioms on top of the concepts and relations.

The acquired ontologies are stored in a relational database. To maximize portability only ANSI-SQL statements are used in the system. All data structures can be serialized into files, different formats like our internal XML-representation OXML, Frame-Logic [15], RDF-Schema [29] and OIL [14] are supported. An F-Logic inference engine described in further detail in [4] can be accessed in OntoEdit.

3 Methodology

The acquisition methodology underlying our approach describing a cyclic acquisition process is depicted in figure 2. The reader may note that this method has shown its usefulness within in our application scenarios, however we do not claim, that this methodology is an optimal mechanism for all purposes, since it is adopted to fit the mentioned *balanced cooperative modeling* paradigm efficiently. It is cyclic to be able to refine and adopt the resulting domain-specific ontology. This approach acknowledges the *evolving nature* of domain-specific ontologies that have adopt to changes in their domain or their application such as described in [5].

The acquisition process starts with the selection of a generic core ontology (cf. subsection 4.1), which has to be converted into our ontology model. Any large generic ontology (like CyC or Dahlgren's ontology), lexical-semantic nets (like WordNet, GermaNet or EuroWordNet) or domain-related ontologies (like TOVE) could start the process. In our case we decided to select GermaNet as the only available German resource that comprises of conceptual as well as lexical resources.

Second, the user must specify which texts should be used in the following steps. This might sound trivial, but this decision strongly influences the results gained from all further steps in the ontology acquisition process and thus the overall output. Both decisions must be regarded as the most influential design decision within our methodology. The next step is to acquire domain-specific concepts from the available resources as the base ontology is (most likely) generic. In our scenario the classification of all newly acquired concepts must also be performed at this point (cf. subsection 4.2). Now the ontology contains domain-specific concepts, but still many generic concepts remain. Therefore the given ontology must be focused to the domain. This happens by removing all generic concepts from the ontology (cf. subsection 4.3). The conceptual structure of the ontology is now established.

Based on this structure the next step acquires non-taxonomic conceptual relations from texts. In addition to the relations provided by the base ontology that survived the focusing step (as their domain/range-concepts still exist) new conceptual relations

[3] A comprehensive description of the ontology engineering system OntoEdit and the underlying methodology is given in [26]

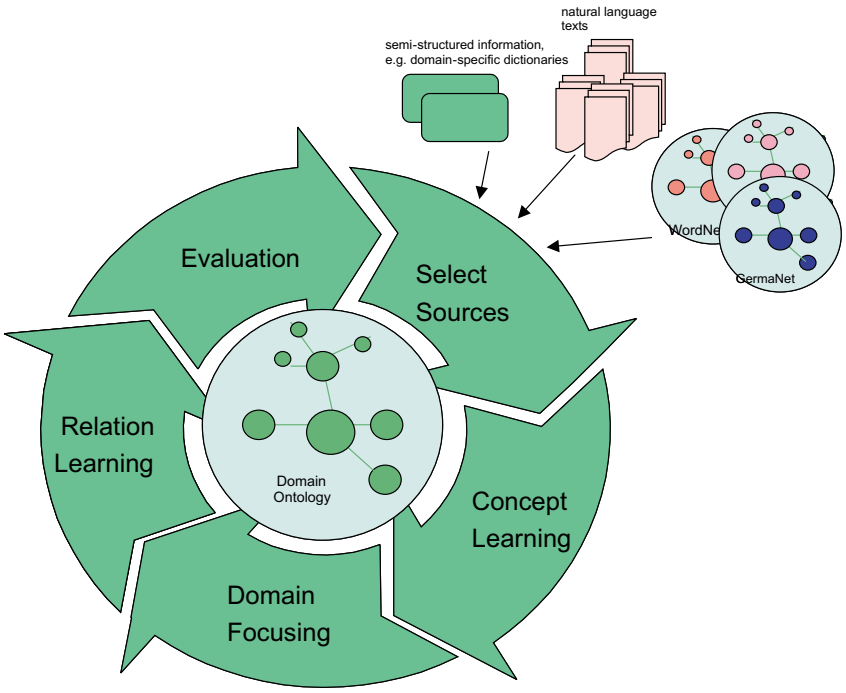

Figure2. Semi-Automatic Ontology Acquisition Process

are induced in the next step by applying learning methods (cf. subsection 4.4) to the selected texts.

4 Acquisition Process

4.1 Base Ontology

We decided to choose a lexical-semantic net for the German language, called GermaNet (cf. [11]) as our base ontology. GermaNet is the German counterpart to the well known WordNet. Presently it builds a lexical semantic network for 16.000 German words, where three different types of word classes are distinguished: nouns, verbs and adjectives. Words are grouped into sets of synomyms so called synsets. Like in WordNet two kinds of relations exist: Lexical relations that hold between words (like antonym) and semantic relations that hold between synsets (like meronym).

Conversion to our ontology primitive. Synsets are regarded as concepts. Therefore only semantic relations can be converted to conceptual relations. Two sematic relations have a special role as they can be used to establish a concept taxonomy. The hypernym relation refers to the synset with a more general meaning. The hyponym relations does the opposite. These relations are not converted to conceptual relations, instead one is used to establish a concept taxonomy. Unfortunally there is no (intended) inheritance within

lexical-semantic nets[4]. As classifying synsets along their hypernym relations ultimately leads to inheritance within our ontology model, not all of the remaining semantic relations can be converted to conceptual relations, but must be identified by the user to be correct after conversion[5]. Actually, we converted each type of relation selected by the user into a generic conceptual relation. All instances of these relations are converted to restrictions in our ontology model (restricting their respective generic relation).

Disambiguation. Every word in a synset is analyzed by the information extraction component to acquire its stem. The stem is assigned to the corresponding concept to get a link to the analyzed texts. This link must be unique for each stem, thus a 1:n relation between a concept and stems is established. Sometimes the same stem is acquired from different synsets. The disambiguation can not be done without using the context a word. As we do not have any relations to do the disambiguation yet, we introduce a new concept to which the ambiguous stem is assigned and make all conflicting synsets subconcepts of the newly introduced concepts. The newly introduced concept is embedded into the taxonomy as a subconcept of the deepest common super concept of all conflicting synsets. The disambiguation must happen later, e.g. it might be possible to disambiguate the users query using the relations between concepts identified in the query.

Resolving problems. Some synsets do not contain any hypernym or hyponym relations. As we strongly intended to have a thorough concept hierarchy, only synsets having at least one hypernym or hyponym relation are converted to concepts. We didn't have access to the final version of GermaNet, this might be a reason that in some (rare) cases synsets (transitively) pointed to themselves through their hyponym relations. In order to leave the taxonomy acyclic we decided to ignore the relations causing a cycle. Unfortunally there were several cycles within the verb classes[6].

4.2 Acquisition of concepts

Getting concepts. In general concepts can be acquired using term frequencies in texts. Terms that are more frequent in a domain-specific corpus than in generic corpora and are not contained in the given ontology should be proposed to the user. In our setting a second kind of input was available. Within Rentenanstalt / Swiss Life a corporate dictionary is maintained to provide that all employees use common translations. We regarded all dictionary entries to be domain-specific concepts. As multi-lingual representations of concepts and relations are available in OntoEdit all translations were entered into the ontology. Figure 3 shows an example. If dictionary entries had multiple headwords (e.g. if acronyms were attached to some terms), headwords were regarded as synonyms and only one concept was created.

[4] This is one of the essential differences to ontologies, where relations must hold for all subconcepts.

[5] Arguably this is true for the meronym and holonym relations.

[6] The only correct interpretation of a cycle is to regard the snysets contaained as synonyms, but this would be modelled in a different manner (one synset), therefore we consider cycles as bugs (and not existant in the final version of GermaNet).

A.D.T.
Automatic Debit Transfer

Electronic service arising from a debit authorization of the Yellow Account holder for a recipient to debit bills that fall due direct from the account.
Cf. also ***direct debit system***.

Figure3. An example entry

Every headword is analyzed by the information extraction component (SMES) to acquire the word stem. This stem is assigned to the newly created concept, if not already existant in the ontology. If this stem exists, we need to find out whether or not the dictionary entry describes the same concept as contained in the ontology.

Resolving conflicts. We apply several heuristics to solve this problem automatically. Table 1 shows the applied heuristics. In general dictionary entries are considered domain-specific and thus more important than existing concepts. The algorithm uses the information included within the dictionary entry and its description to find out whether the entry denotes the existing concept or induces a new concept.

Property	**Automatic resolution**
Word is acronym	Remove stem reference in ontology
Dictionary entry has no description	Do not import the entry and keep concept in ontology
Dictionary entry and ontology entry have a common super concept	Do not import the entry and keep concept in ontology
else	ask the user to resolve the conflict

Table1. Dictionary: Resolution for stem conflicts

First, the algorithm checks whether the conflicting dictionary head word denotes an acronym (e.g. *ALE* is an acronym for *unemployment benefits* in German. Unfortunately the stem reference contained in the ontology points to the concept *ale*, which is a subconcept of *alcoholic beverage*), in this case the stem reference is re-assigned to the dictionary concept. If this doesn't help, the algorithm checks whether further information is contained in the dictionary description by trying to find the super concept using the taxonomy acquisition method explained in the next paragraph. If a found super concept is also a super concept of the concept in the ontology, the dictionary entry and the concept in the ontology are considered equal. If no descriptions are contained in the dictionary entry and the entry is not an acronym , the concept in the ontology is kept. Last but not least, if none of heuristics could be applied, the user is asked to resolve the conflict.

Getting the taxonomy We used several heuristics to acquire the concept classification required to build a taxonomy. Texts (and the descriptions of all dictionary entries) are analyzed using the information extraction component. Several heuristics are applied to its output. The first heuristic used is applying pattern matching to texts. This heuristic

is motivated by [12], who brought up the idea, that certain patterns in texts induce a hyponym relation between words. This idea - that was also successfully applied in [20] - worked quite well due to the fact that the information extraction component supplies regular output, figure 4 depicts a very successful pattern.

The second heuristic used deals with compounds, that are very frequent in German, for example "Arbeitslosenentschädigung" - in English "unemployment benefits" - is a compound. The information extraction component can decompose compounds, thus supplying parts of compounds. Our heuristic treats the last part of a compound delivered by the information extraction system as a hypernym and suggests the concept retrieved using the supplied stem as a superconcept. The third heuristic used deals with phrasal compounds like "Automatic Debit Transfer". The last noun in a noun phrase is determined and refers a superconcept by its stem. Before the ISA-relations retrieved by the heuristics are presented to the user, several consistency checks are performed. First, all stems that do not refer to a concept in the ontology are determined and suggested to the user for assignment to a concept[7]. Second, if several superconcepts were found in the same path, only the deepest concept is presented to the user.

Pattern:
1. *lexicon entry* :: (NP_1, NP_2, NP_i, *and / or* NP_n)
2. for all NP_i, $1 <= i <= n$ hypernym(NP_i, *lexicon entry*)

Result: hypernym("electronic service", "A.D.T.")

Figure4. Pattern Definition

4.3 Removal of concepts

We motivated the removal of generic concepts in section 3. In order to prune domain-unspecific concepts, concept frequencies are determined from the selected domain-specific documents (see [24]). Concept frequencies are also determined from a second corpus that contains generic documents (as found in reference corpora like CELEX). We used the publicly available archive of a well-known German newspaper (http://www.taz.de/) as generic corpus. All concept frequencies are propagated to superconcepts, by summarizing the frequencies of subconcepts. Then the frequencies of both corpora are compared using a measure selected by the user. The user can choose from the well known standard measures (used widely in the information retrieval community) TF (term frequency) and TFIDF (term frequency - inverted document frequency). TFIDF attaches a term-weighting factor to the original TF, which punishes all terms that are frequent in all documents, using a collection frequency. All existing concepts that are more frequent [8] within the domain-specific corpus than in the generic corpus remain in the ontology. The user can also specify whether or not concepts, that are not contained in the domain-specific and the generic corpus, should be pruned.

As our ontology is intended to be used to work with texts in general (either in a retrieval or in a semantic annotation scenario) we have to minimize the loss of refer-

[7] The user can create a new concept if no concept in the ontology has the intended meaning of the stem.

[8] A factor has to be provided by the user.

ences from words to concepts. For this reason, we do not delete stem references but move them to the closest superconcept that remains in the ontology. Only if multiple superconcepts in distinct paths remain, the stem reference is deleted. For example, if chair" is pruned from the ontology, we might move the stem reference to "furniture" being the closest superconcept that remains in the ontology.

4.4 Acquisition of Conceptual Relations

This approach is founded on the idea that frequent couplings of concepts in sentences can be regarded as relevant relations between concepts. We adopted an algorithm based on association rules (see [25]) to find frequent correlations between concepts. Linguistically processed texts as input, where coupling of concepts within sentences are retrieved, are processed by our algorithm. Consult [28,16] for a detailed description of this approach.

Two measures denote the statistical data derived by the algorithm: *Support* measures the quota of a specific coupling within the total number of couplings. *Confidence* denotes the part of all couplings supporting both domain and range concepts within the number of couplings that support the same domain concept. The retrieved measures are propagated to super concepts using the background knowledge provided by the taxonomy. This strategy is used to emphasize the couplings in higher levels of the taxonomy.

For instance, the linguistic processing may find that the word "policy" frequently co-occurs with each of the words "policy owner" and "insurance salesman". From this statistical linguistic data our approach derives correlations at the conceptual level, viz. between the concept Policy and the concepts, PolicyOwner and InsuranceSalesman. 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 Policy and Person. 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 Policy and Person may be proposed for inclusion in the ontology.

Results are presented to the user, if the measures of a coupling satisfy specific minimum values provided by the user. Also, the input structures can be restricted to a set of certain concepts (whereas at least one element of every coupling must be in the given set) to be able to do a more focused way of relation acquisition.

We have to stress that this method can only be used to retrieve suggestions that are presented to the user. Manual labour is still needed to select and name the relations. To simplify user access results are conveniently displayed in the common result structure. The correctness towards the inheritance property of the taxonomy is automatically determined.

5 Results

We here only present partial results at the moment, since our work is still ongoing. The overall results for each step of the acquisition process are shown in table 2 using some statistical values about the ontology (namely the number of concepts, the number of relations, the average and maximum depth of the concept taxonomy and the number

of domain lexicon entries). We did not spend any manual engineering effort yet. The acquisition of concepts was also limited to the dictionary.

Acquisition step	$\|C\|$	$\|R\|$	$\emptyset$ Depth Tax.	Max. Depth Tax.	|DLex Entr.|
Base Ontology	19.404	2.713	7.19	18	20.617
Concept Learning	20.412	2.713	7.17	18	21.987
Concept Removal	4.358	673	6.28	17	13.628

Table2. Ontology statistics after each step of the acquisition process

Base ontology. We converted only one nouns from the GermaNet lexical-semantic net. Our version contained 18451 synsets for the noun class. 18058 synsets have been converted, since we didn't consider synsets that were not embedded in the taxonomy (by having neither hypernyms nor hyponyms). 1346 new concepts were introduced due to our disambiguation strategy. Therefore 19404 concepts were created. We converted only two semantic relations, namely the meronym and holonym relations. Converting these relations led to 2713 relations.

Concept acquisition. The dictionary contained 1116 entries, 94 stems of these entries were already contained in the ontology. 12 stem conflicts were within the dictionary [9]. The automatic resolution resolved 26% of the problems. In 50% of the resolved cases acronyms where involved, all other entries where found to be equal with the concepts in the ontology, as the same super concepts were found using the ISA-Heuristics. Manual resolution of the remaining 74% of the problems found most of the entries to be equal of the concepts in the ontology. The ISA-Heurisics found 1336 is-a relations, where 52% were found using the compound heuristics. 48% of these results were found using the pattern heuristic. 427 (32%) of the results were found to be wrong by user evaluation. Removing these results leads to the lack of 215 superconcepts, thus 15% of all dictionary entries must be aligned into the taxonomy manually.

Concept removal. The selected domain-specific corpus comprised of 1153 intranet documents. 6540 terms were extracted from this corpus. The generic corpus comprised of 255 documents from the TAZ newspaper archive. This corpus contained 6154 terms. Unfortunately only 1881 terms (30%) from the domain-specific corpus referenced any concepts in the ontology. Therefore, at the maximum only these concepts and their superconcepts can survive the pruning step. Using a ratio of 1.0 selects all of these terms independent of the measure selected. The pruning of the ontology leaves 4358 concepts. This result emphasizes the need to retrieve concepts from the selected corpus, since 70% of the terms in the corpus did not reference any concepts in the ontology.

Acquisition of relations. This learning method has not been thoroughly evaluated with our corpus yet. Results regarding the acquistion of relations using our statistical

[9] Due to the fact that the information extraction component retrieves the same stem for strings like "CH/IFUE1" and "CH/IFUE2"

approach in a different domain (tourism) have been presented in [16]. Best results were reached using a minimum support value of 0.04 and a minimum confidence of 0.01. 98 relations were discovered using an ontology that contained 284 concepts and 88 conceptual relations. 11% of the discovered relations were already modeled before, thus 13% of the hand-modeled relations were discovered by the learning algorithm.

6 Conclusions & Further Work

In this paper we have described our recent and ongoing work in semi-automatic ontology acquisition from a corporate intranet. Based on our comprehensive architecture a new approach for supporting the overall process of engineering ontologies from text is described. It is mainly based on a given core ontology, which is extended with domain specific concepts. The resulting ontology is pruned and restricted to a specific application using a corpus-based mechanism for ontology pruning. On top of the ontology two approaches supporting the difficult task of determining non-taxonomic conceptual relationships are applied.

In the future much work remains to be done. First, several techniques for evaluating the acquired ontology have to be developed. In our scenario we will apply ontology cross comparison techniques such as described in [16]. Additionally, applying the ontology on top of the intranet documents (e.g. a information retrieval scenario, a semantic document annotation scenario such as described in [5]) will allow us an application-specific evaluation of the ontology using standard measures such as precision and recall. Second, our approach for multi-strategy learning is still in an early stage. We will have to elaborate how the results of different learning algorithms will have to be assessed and combined in the multi-strategy learning set. Nevertheless, an approach combing different resources on which different techniques are applied, seems promising for supporting the complex task of ontology learning from text.

Acknowledgements: This work has been partially found by the European Union and the Swiss Government under the contract-no "BBW Nr.99.0174" as part of the European commision Research Project "IST-1999-10132" (On-to-Knowledge). We thank the DFKI, language technology group, in particular Günter Neumann, who generously supported us in using their SMES system.

References

1. D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. FASTUS: A finite state processor for information extraction from real world text. In *IJCAI-93: 13International Joint Conference on Artificial Intelligence. Chambery, France, August 28 - September 3, 1993*, Chambery, France, August 1993.
2. A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.
3. Roy J. Byrd and Yael Ravin. Identifying and extracting relations from text. In *NLDB'99 — 4th International Conference on Applications of Natural Language to Information Systems*, 1999.

4. S. Decker. On domain-specific declarative knowledge representation and database languages. In *Proc. of the 5th Knowledge Representation meets Databases Workshop (KRDB98)*, pages 9.1–9.7, 1998.
5. M. Erdmann, A. Maedche, H.-P. Schnurr, and Steffen Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *P. Buitelaar & K. Hasida (eds). Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg, August 2000.
6. David Faure and Claire Nedellec. Knowledge acquisition of predicate-argument structures from technical texts using machine learning. In *Proc. of Current Developments in Knowledge Acquisition, EKAW-99*, 1999.
7. D. Fensel, F. van Harmelen, H. Akkermans, M. Klein, J. Broekstra, C. Fluyt, J. van der Meer, H.-P. Schnurr, R. Studer, J. Davies, J. Hughes, U. Krohn, R. Engels, B. Bremdahl, F. Ygge, U. Reimer, and I. Horrocks. Ontoknowledge: Ontology-based tools for knowledge management. In *Proceedings of the eBusiness and eWork 2000 Conferencee (EMMSEC 2000)*, Madrid, Spain, To appear October 2000.
8. E. Grosso, H. Eriksson, R. W. Fergerson, S. W. Tu, and M. M. Musen. Knowledge modeling at the millennium — the design and evolution of Protégé-2000. In *Proc. the 12th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW'99), Banff, Canada, October 1999*, 1999.
9. T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL-93-04, Stanford Knowledge Systems Laboratory (KSL), Stanford University, 1993.
10. Udo Hahn and Klemens Schnattinger. Towards text knowledge engineering. In *AAAI '98 - Proceedings of the 15th National Conference on Artificial Intelligence. Madison, Wisconsin, July 26-30, 1998*, pages 129–144, Cambridge/Menlo Park, 1998. MIT Press/AAAI Press.
11. 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.
12. M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France*, 1992.
13. I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, and R. Studer. The ontology inference layer oil, on-to-knowledge eu-ist-10132 project deliverable no. otk-d1. Technical report, Free University Amsterdam, Division of Mathematics and Computer Science, Amsterdam, NL, 2000.
14. I. Horrocks et.al. The ontology interchange language oil: The grease between ontologies. Technical report, Dep. of Computer Science, Univ. of Manchester, UK/ Vrije Universiteit Amsterdam, NL/ AIdministrator, Nederland B.V./ AIFB, Univ. of Karlsruhe, DE, 2000. http://www.cs.vu.nl/˜dieter/oil/.
15. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42, 1995.
16. A. Maedche and S. Staab. Discovering conceptual relations from text. In *Proceedings of ECAI-2000*. IOS Press, Amsterdam, 2000.
17. A. Maedche and S. Staab. Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering. Chicago, USA*. KSI, 2000.
18. R. Michalski and K. Kaufmann. Data mining and knowledge discovery: A review of issues and multistrategy approach. In *Machine Learning and Data Mining Methods and Applications*. John Wiley, England, 1998.
19. K. Morik. Balanced cooperative modeling. *Machine Learning*, 11:217–235, 1993.
20. 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.

21. 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.
22. G. Neumann, C. Braun, and J. Piskorski. A divide-and-conquer strategy for shallow parsing of german free texts. In *Proceedings of ANLP-2000, Seattle, Washington*, 2000.
23. P. Resnik. *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, University of Pennsylania, 1993.
24. G. Salton. *Automatic Text Processing*. Addison-Wesley, 1988.
25. R. Skrikant and R. Agrawal. Mining generalized association rules. In *Proceedings of VLDB 1995*, pages 407–419, 1995.
26. 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.
27. M. Uschold. Building ontologies: Towards a unified methodology. In *Expert Systems 96, Cambridge*, Dezember 1996.
28. Raphael Volz. Discovering conceptual relations from text. Studienarbeit, University of Karlsruhe (TH), Karlsruhe - Germany, 2000. in German.
29. W3C. Rdf schema specification. http://www.w3.org/TR/PR-rdf-schema/, 1999.
30. 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.