On2L - A Framework for Incremental Ontology Learning in Spoken Dialog Systems

Berenike Loos
European Media Laboratory GmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
berenike.loos@eml-d.villa-bosch.de

Abstract

An open-domain spoken dialog system has to deal with the challenge of lacking lexical as well as conceptual knowledge. As the real world is constantly changing, it is not possible to store all necessary knowledge beforehand. Therefore, this knowledge has to be acquired during the run time of the system, with the help of the out-of-vocabulary information of a speech recognizer. As every word can have various meanings depending on the context in which it is uttered, additional context information is taken into account, when searching for the meaning of such a word.

In this paper, I will present the incremental ontology learning framework On2L. The defined tasks for the framework are: the hypernym extraction from Internet texts for unknown terms delivered by the speech recognizer; the mapping of those and their hypernyms into ontological concepts and instances; and the following integration of them into the system's ontology.

1 Introduction

A computer system, which has to understand and generate natural language, needs knowledge about the real world. As the manual modeling and maintenance of those knowledge structures, i.e. ontologies, are both time and cost consuming, there exists a demand to build and populate them automatically or at least semi automatically. This is possible by analyzing unstructured, semi-structured or fully structured data by various linguistic as well as statistical means and by converting the results into an ontological form.

In an open-domain spoken dialog system the automatic learning of ontological concepts and corresponding relations between them is essential, as a complete manual modeling of them is neither practicable nor feasible as the real world and its objects, models and processes are constantly changing and so are their denotations.

This work assumes that a viable approach to this challenging problem is to learn ontological concepts and relations relevant for a certain user - and only those - incrementally, i.e. at the time of the user's inquiry. Hypernyms[1] of terms that are not part of the speech recognizer lexicon, i.e. out-of-vocabulary (OOV) terms, and hence lacking any mapping to the employed knowledge representation of the language understanding component, should be found in texts from the Internet. That is the starting point of the proposed ontology learning framework *On2L* (On-line Ontology Learning). With the found hypernym On2L can assign the place in the system's ontology to add the unknown term.

So far the work described herein refers to the German language only. In a later step, the goal is to optimize it for English as well.

2 Natural Language and Ontology Learning

Before describing the actual ontology learning process it is important to make a clear distinction between the two fields involved: this is on the one hand natural language and on the other hand ontological knowledge.

As the Internet is a vast resource of up-to-date

[1] According to Lyons (1977) hyponymy is the relation which holds between a more specific lexeme (i.e. a hyponym) and a more general one (i.e. a hypernym). E.g. animal is a hypernym of cat.

Proceedings of the COLING/ACL 2006 Student Research Workshop, pages 61–66,
Sydney, July 2006. ©2006 Association for Computational Linguistics

information, On2L employs it to search for OOV terms and their corresponding hypernyms. The natural language texts are rich in terms, which can be used as labels of concepts in the ontology and rich in semantic relations, which can be used as ontological relations.

The two areas which are working on similar topics but are using different terminology need to be distinguished, so that the extraction of semantic information from natural language is separated from the process of integrating this knowledge into an ontology.

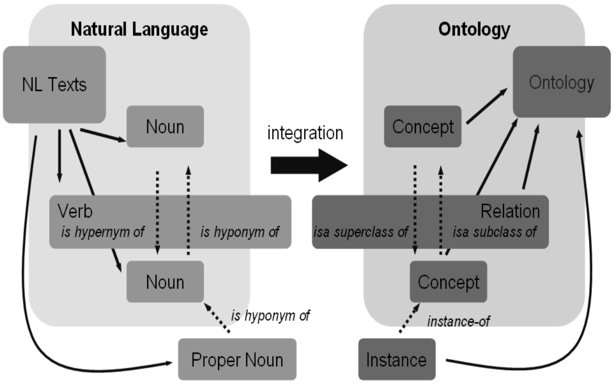

Figure 1: Natural Language and Ontology Learning

Figure 1 shows the process of ontology learning from natural language text. On the left side natural language lexemes are extracted. During a transformation process nouns, verbs and proper nouns are converted into concepts, relations and instances of an ontology[2].

3 Related Work

The idea of acquiring knowledge exactly at the time it is needed is new and became extremely useful with the emergence of open-domain dialog systems. Before that, more or less complete ontologies could be modeled for the few domains covered by a dialog system. Nonetheless, many ontology learning frameworks exist, which alleviate the work of an ontology engineer to construct knowledge manually, e.g. ASIUM (Faure and Nedellec, 1999), which helps an expert in acquiring knowledge from technical text using syntactic analysis for the extraction, a semantic similarity measure and a clustering algorithm for the conceptualization. OntoLearn (Missikoff et al., 2002) uses specialized web site texts as a corpus to extract terminology, which is filtered by statistical techniques and then used to create a domain concept forest with the help of a semantic interpretation and the detection of taxonomic and similarity relations. KAON Text-To-Onto (Maedche and Staab, 2004) applies text mining algorithms for English and German texts to semi-automatically create an ontology, which includes algorithms for term extraction, for concept association extraction and for ontology pruning.

Pattern-based approaches to extract hyponym/hypernym relationships range from hand-crafted lexico-syntactic patterns (Hearst, 1992) to the automatic discovery of such patterns by e.g. a minimal edit distance algorithm (Pantel et al., 2004).

The SmartWeb Project into which On2L will be integrated as well, aims at constructing an open-domain spoken dialog system (Wahlster, 2004) and includes different techniques to learn ontological knowledge for the system's ontology. Those methods work offline and not at the time of the user's inquiry in contrast to On2L:

C-PANKOW (Cimiano et al., 2005) puts a named entity into several linguistic patterns that convey competing semantic meanings. The patterns, which can be matched most often on the web indicate the meaning of the named entity.

RelExt (Schutz and Buitelaar, 2005) automatically identifies highly relevant pairs of concepts connected by a relation over concepts from an existing ontology. It works by extracting verbs and their grammatical arguments from a domain-specific text collection and computing corresponding relations through a combination of linguistic and statistical processing.

4 The ontology learning framework

The task of the ontology learning framework On2L is to acquire knowledge at run time. As On2L will be integrated into the open-domain dialog system Smartweb (Wahlster, 2004), it will be not only useful for extending the ontology of the system, but to make the dialog more natural and therefore user-friendly.

Natural language utterances processed by an open-domain spoken dialog system may contain words or parts of words which are not recognized by the speech recognizer, as they are not contained

[2] In our definition of the term *ontology* not only concepts and relations are included but also instances of the real world.

in the recognizer lexicon. The words not contained are most likely not represented in the word-to-concept lexicon as well[3]. In the presented ontology learning framework On2L the corresponding concepts of those terms are subject to a search on the Internet. For instance, the unknown term *Auerstein* would be searched on the Internet (with the help of a search engine like Google). By applying natural language patterns and statistical methods possible hypernyms of the term can be extracted and the corresponding concept in the ontology of the complete dialog system can be found. This process is described in Section 4.5.

As a term often has more than one meaning depending on the context in which it is uttered, some information about this context is added for the search[4] as shown in Section 4.4.

Figure 2 shows the life cycle of the On2L framework. In the middle of the diagram the question example by a supposed user is: *How do I get to the Auerstein?* The lighter fields in the figure mark components of the dialog system, which are only utilized by On2L, whereas the darker fields are especially built to complete the ontology learning task.

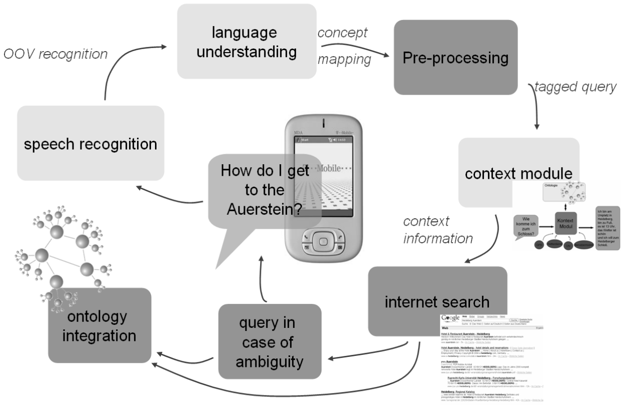

Figure 2: The On2L Life Cycle

The sequential steps shown in Figure 2 are described in more detail in the following paragraphs starting with the processing of the user's utterance by the speech recognizer.

4.1 Speech Recognition

The speech recognizer classifies all words of the user's utterance not found in the lexicon as out-of-vocabulary (OOV). That means an automatic speech recognition (ASR) system has to process words, which are not in the lexicon of the speech recognizer (Klakow et al., 2004). A solution for a phoneme-based recognition is the establishment of corresponding best rated grapheme-chain hypotheses (Gallwitz, 2002). These grapheme-chains are constructed with the help of statistical methods to predict the most likely grapheme order of a word, not found in the lexicon. Those chains are then used for a search on the Internet in the final version of On2L. To evaluate the framework itself adequately so far only a set of correctly written terms is subject to search.

4.2 Language Understanding

In this step of the dialog system, all correctly recognized terms of the user utterance are mapped to concepts with the help of a word-to-concept lexicon. Such a lexicon assigns corresponding natural language terms to all concepts of an ontology. This is not only a necessary step for the dialog system, but can assist the ontology learning framework in a possibly needed semantic disambiguation of the OOV term.

Furthermore the information of the concepts of the other terms of the utterance can help to evaluate results: when there are more than one concept proposal for an instance (i.e. on the linguistic side a proper noun like *Auerstein*) found in the system's ontology, the semantic distance between each proposed concept and the other concepts of the user's question can be calculated[5].

4.3 Preprocessing

A statistical part-of-speech tagging method decides on the most probable part-of-speech of the whole utterance with the help of the sentence context of the question. In the On2L framework we used the language independent tagger qtag[6], which we trained with the hand-tagged German corpus NEGRA 2[7].

[3] In case the speech recognizer of the system and the word-to-concept lexicon are consistent.

[4] Of course, even in the same context a term can have more than one meaning as discussed in Section 4.6.

[5] E.g. with the single-source shortest path algorithm of Dijkstra (Cormen et al., 2001).

[6] qtag exists as a downloadable JAR file and can therefore be integrated into a platform independent JAVA program. For more information, see http://www.english.bham.ac.uk/staff/omason/software/qtag.html (last access: 21st February 2006).

[7] The NEGRA corpus version 2 consists of 355,096 tokens (20,602 sentences) of German newspaper text, taken from the Frankfurter Rundschau. For more information visit: http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html (last access: 21st February 2006).

With the help of this information, the part-of-speech of the hypernym of the OVV term can be predicted. Furthermore, the verb(s) of the utterance can anticipate possible semantic relations for the concept or instance to be integrated into the ontology.

4.4 Context Module

To understand the user in an open-domain dialog system it is important to know the extra-linguistic context of the utterances. Therefore a context module is applied in the system, which can give information on the discourse domain, day and time, current weather conditions and location of the user. This information is important for On2L as well. Here we make use of the location of the user and the discourse domain so far, as this information is most fruitful for a more specific search on the Internet. The location is delivered by a GPS component and the discourse domain is detected with the help of the pragmatic ontology PrOnto ((Porzel et al., 2006)). Of course, the discourse domain can only be detected for domains modeled already in the knowledge base (Rueggenmann and Gurevych, 2004).

The next section will show the application of the context terms in more detail.

4.5 Hypernym extraction from the Internet

We apply the OOV term from the speech recognizer as well as a context term for the search of the most likely hypernym on the Internet.

For testing reasons a list of possible queries was generated. Here are some examples to give an idea:

(1) Auerstein – Heidelberg

(2) Michael Ballack – SportsDiscourse

(3) Lord of the Rings – CinemaDiscourse

On the left side of the examples 1 to 3 is the OOV term and on the right side the corresponding context term as generated by the context module. For searching, the part "Discourse" is pruned.

The reason to lay the main focus of the evaluation searches on proper nouns is, that those are most likely not in the recognizer lexicon and not as instances in the system's ontology.

4.5.1 Global versus Local OOVs

To optimize results we make a distinction between global OOVs and local OOVs.

In the case of generally familiar proper nouns like stars, hotel chains or movies (so to say global OOVs), a search on Wikipedia can be quite successful.

In the case of proper nouns, only common in a certain country region, like Auerstein (Restaurant), Bierbrezel (Pub) and Lux (Cinema), which are local OOVs, a search with Wikipedia is generally not fruitful. Therefore it is searched with the help of the Google API.

As one can not know the kind of OOV beforehand, the Wikipedia search is started before the Google search. If no results are produced, the Google search will deliver them hopefully. If results are found, Google search will be used to test those.

4.5.2 Wikipedia Search

The structure of Wikipedia[8] entries is preassigned. That means, the program can know, where to find the most suitable information beforehand. In the case of finding hypernyms the first sentence in the encyclopedia description is most useful. To give an example, here is the first sentence for the search entry *Michael Ballack*:

(4) *Michael Ballack* (born September 26, 1976 in Grlitz, then East Germany) IS A German **football player**.

With the help of lexico-syntactic patterns, the hypernym can be extracted. Those so-called Hearst patterns (Hearst, 1992) occur frequently in lexicons for describing a term. In example 4 the pattern *X is a Y* would be matched and the hypernym *football player*[9] of the term *Michael Ballack* could be extracted.

4.5.3 Google Search

The search parameters in the Google API can be adjusted for the corresponding search task. The tasks we used for our framework are a search in the titles of the web pages and a search in the text of the web pages.

Adjusting the Google parameters The assumption was, that depending on the task the Google parameters should be adjusted. Four parameters were tested with the two tasks (Title and

[8] Wikipedia is a free encyclopedia, which is editable on the Internet: www.wikipedia.org (last access: 22nd February 2006)

[9] In German compounds generally consist of only one word, therefore it is easier to extract them than in the case of English ones.

Page Search, as described in the next paragraphs) and a combination thereof. The parameter *default* is used, when no other parameters are assigned; *intitle* is set, in case the search term should be found in the title of the returned pages; *allintext*, when the search term should be found in the text of the pages; and *inurl*, when the search term should be found in the URL.

In Figure 3 the outcome of the evaluation is shown. The evaluation was done by students, who scored the titles and pages with 1, when a possible hypernym could be found and 0 if not. Surprisingly, the default value delivered the best results for all tasks, followed by the allintext parameter.

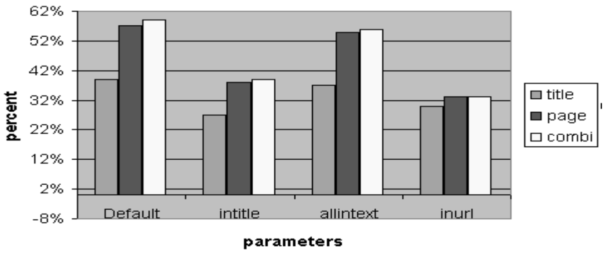

Figure 3: Evaluation of the Google parameters

Title Search To search only in the titles of the web pages has the advantage, that results can be generated relatively fast. This is important as time is a relevant factor in spoken dialog systems. As the titles often contain the hypernym but do not consist of a full sentence, Hearst patterns cannot be found. Therefore, an algorithm was implemented, which searches for nouns in the title, extracts them and counts the occurrences. The noun most frequently found in all the titles delivered by Google is regarded as the hypernym. For the counting we applied stemming and clustering algorithms to group similar terms.

Page Search For Page Search Hearst patterns as in Wikipedia Search were applied. In contrast to encyclopedia entries the recall of those patterns was not so high in the texts from the web pages.

Thus, we searched in the text surrounding of the searched term for nouns. Equally to Title Search we counted the occurrence of nouns. Different evaluation steps showed, that the window size of four words in front and after the term is most successful.

With the help of machine learning algorithms from the WEKA[10] library we did a text mining to ameliorate the results as shown in Faulhaber et al. (2006).

4.5.4 Results

Of all 100 evaluated pages for Google parameters only about 60 texts and about 40 titles contained possible hypernyms (as shown in Figure 3). This result is important for the evaluation of the task algorithms as well. The outcome of the evaluation setup was nearly the same: 38 % precicion for Title Search and about 58 % for Page Search (see Faulhaber (2006)). These scores where evaluated with the help of forms asking students: *Is X a hypernym of Y?*.

4.6 Disambiguation by the user

In some cases two or more hypernyms are scored with the same – or quite similar – weights. An obvious reason is, that the term in question has more than one meaning in the same context. Here, only a further inquiry to the user can help to disambiguate the OOV term. In the example from the beginning a question like "Did you mean the hotel or the restaurant?" could be posed. Even though the system would show the user that it did not perfectly understand him/her, the user might be more contributory than in a question like "What did you mean?". The former question could be posed by a person familiar with the place, to disambiguate the question of someone in search for *Auerstein* as well and would therefore mirror a human-human dialog leading to more natural dialogs with the machine.

4.7 Integration into the ontology

The foundational ontology (Cimiano et al., 2004) integrated into the dialog system Smartweb is based on the highly axiomatized Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [11]. It features various extensions called *modules*, e.g. *Descriptions & Situations* (Gangemi and Mika, 2003). Additional to the foundational ontology a domain-independent layer is included which consists of a range of branches from the less axiomatic SUMO (Suggested Upper Merged Ontology (Niles and Pease, 2001)), which is known for its intuitive and comprehensible structure. Currently, the dialog system features several domain

[10] http://www.cs.waikato.ac.nz/ml/weka (last access: 21st February 2006).

[11] More information on this descriptive and reductionistic approach is found on the WonderWeb Project Homepage: wonderweb.semanticweb.org.

ontologies, i.e. a SportEvent-, a Navigation-, a WebCam-, a Media-, and a Discourse-Ontology.

According to this, it is possible that in some cases there exists the corresponding concept to a hypernym. This can be found out with the help of a so-called term widening. The concept labels in the SmartWeb Ontology are generally English terms. Therefore the found German hypernym has to be translated into English. An English thesaurus is used to increase the chance of finding the right label in the ontology.

5 Future Work

The work described here is still in process and not evaluated in detail so far. Therefore, our goal is to establish a task-oriented evaluation setup and to ameliorate the results with various techniques.

As natural language texts are not only rich in hierarchical relations but in other semantic relations as well, it is advantageous to extend the ontology by those relations.

As user contexts are an important part of a dialog system, we are planning to learn new user contexts, which can be represented in the ontology by the DOLCE module Descriptions and Situations.

Furthermore our goal is, to integrate the ontology learning framework into the open-domain spoken dialog system Smartweb.

References

Philipp Cimiano, Andreas Eberhart, Daniel Hitzler, Pascal Oberle, Steffen Staab, and Rudi Studer. 2004. The smartweb foundational ontology. *SmartWeb Project Report*.

Philipp Cimiano, Günter Ladwig, and Steffen Staab. 2005. Gimme' the context: Context-driven automatic semantic annotation with c-pankow. In *Proceedings of the 14th World Wide Web Conference*. ACM Press.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. Section 24.3: Dijkstra's algorithm. In *Introduction to Algorithms, Second Edition*, pages 595–601. MIT Press and McGraw-Hill.

Arndt Faulhaber, Berenike Loos, Robert Porzel, and Rainer Malaka. 2006. Towards understanding the unknown: Open-class named entity classification in multiple domains. In *Proceedings of the Ontolex Workshop at LREC*. Genoa, Italy.

David Faure and Claire Nedellec. 1999. Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system asium. In *EKAW '99: Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management*, London, UK. Springer-Verlag.

Florian Gallwitz. 2002. *Integrated Stochastic Models for Spontaneous Speech Recognition*. Logos, Berlin.

Aldo Gangemi and Peter Mika. 2003. Understanding the semantic web through descriptions and situations. In *Proceedings of the ODBASE Conference*. Springer.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, Nantes, France.

Dietrich Klakow, Georg Rose, and Xavier Aubert. 2004. Oov-detection in a large vocabulary system using automatically defined word-fragments as filler. In *Proceedings of EUROSPEECH'99*, Budapest, Hungary.

John Lyons. 1977. *Semantics*. University Press, Cambridge, MA.

Alexander Maedche and Steffen Staab. 2004. Ontology learning. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems. Springer.

Michele Missikoff, Roberto Navigli, and Paola Velardi. 2002. Integrated approach to web ontology learning and engineering. In *IEEE Computer - November*.

Ian Niles and Adam Pease. 2001. Towards a standard upper ontology. In Chris Welty and Barry Smith, editors, *Workshop on Ontology Management*, Ogunquit, Maine. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001).

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale semantic acquisition. In *Proceedings of Coling*, Geneva, Switzerland. COLING.

Robert Porzel, Hans-Peter Zorn, Berenike Loos, and Rainer Malaka. 2006. Towards a separation of pragmatic knowledge and contextual information. In *Proceedings of ECAI-06 Workshop on Contexts and Ontologies*, Lago di Garda, Italy.

Klaus Rueggenmann and Iryna Gurevych. 2004. Assigning domains to speech recognition hypotheses. In *Proceedings of HLT-NAACL Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Knowledge for Speech Processing*. Boston, USA.

Alexander Schutz and Paul Buitelaar. 2005. Relext: A tool for relation extraction in ontology extension. In *Proceedings of the 4th International Semantic Web Conference*. Galway, Ireland.

Wolfgang Wahlster. 2004. SmartWeb: Mobile applications of the semantic web. In *Proceedings of Informatik*, Ulm, Germany.