Automatic Acquisition of Sense Examples using ExRetriever

Montse Cuadros[1] and Jordi Atserias[1] and Mauro Castillo[1] and German Rigau[2]

[1] TALP Research Center. UPC
{cuadros,castillo,batalla}@lsi.upc.es
[2] IXA Research Group. UPV/EHU
rigau@si.ehu.es

Abstract. A promising research line for word sense disambiguation (WSD) focuses on the use of supervised machine learning techniques. One of the drawbacks of using such techniques is that they requires previously sense annotated data. This paper presents ExRetriever, a new software tool for automatically acquiring large sets of sense tagged examples from large collections of text (e.g. the Web). ExRetriever exploits large-scale knowledge bases (e.g., WordNet) to build complex queries, each of them characterising particular senses of a word. These examples can be used as training instances for supervised WSD algorithms.

1 Introduction

A promising current line of research of WSD uses semantically annotated corpora to train Machine Learning algorithms to decide which word sense to choose in which contexts. These approaches are called "supervised" because they learn from previously sense annotated data.

Supervised Machine Learning algorithms use semantically annotated corpora to induce classification models for deciding which is the appropriate word sense for each particular context. Compilation of corpora for training and testing such systems require a large human effort since all the words in these annotated corpora have to be manually tagged by lexicographers with semantic classes taken from a particular lexical semantic resource, most commonly WordNet. Supervised methods suffer from the lack of widely available semantically tagged corpora, from which to construct really broad coverage systems. This extremely high overhead for supervision (all words, all languages) explain why supervised methods have been seriously questioned.

As a possible solution, some recent work is focusing on reducing the acquisition cost and the need for supervision in corpus-based methods for WSD. For instance, [1], [2] and [3] automatically generate arbitrarily large corpora for unsupervised WSD training, using the knowledge contained in WordNet to formulate search engine queries over large text collections or the Web.

2 Automatic Acquisition of Examples for WSD

The work of Leacock et al. [1] using AutoTrain collected monosemous relatives. The sampling process retrieves the ”closest” relatives first. The quality of the acquired data was evaluated indirectly comparing the results of a WSD system for 14 nouns when trained on monosemous relatives and on manually tagged training materials. The result of this experiment was that some words could be automatically tagged with nearly human rates of success, but there were other words for which automatic tagging was not worthy.

Mihalcea and Moldovan [2] try to overcome these limitations (1) by using the word definitions provided by glosses and (2) by using the Web as a very large corpus. In this case, they use Altavista to create complex search queries using boolean operators for increasing the quality of the information retrieved. Their approach was tested on 20 polysemous words giving an accuracy of 91%. Using this method for these words, they obtained thirty times more examples than appearing in SemCor.

Agirre and Martinez [3] implemented the previously described method of Mihalcea and Moldovan to obtain training data for 13 words, and tested on examples from SemCor. Only a few words get better results than random and for a particular word the error rate reached 100%.

Agirre and Martínez suggest that one possible explanation of this apparent disagreement could be that the acquired examples, being correct on themselves, provide systematically misleading features (for instance, as suggested by [1] when using a large set of local closed-class and part-of-speech features). Besides, all words were trained with equal number of examples.

In order to test the feasibility of this approach, the Meaning project[3] has developed and released a new tool: the first version of ExRetriever, a flexible system to perform sense queries on large corpora. ExRetriever characterizes automatically each synset of a word as a query (using mainly: synonyms, hyponyms and the words of the definitions); and then, uses these queries to obtain sense examples (sentences) automatically from a large text collection. The current implementation of ExRetriever accesses directly the content of the Multilingual Central Repository (Mcr) [4] of the Meaning project. The system is using also SWISH-E[4] to index large collections of text such as SemCor or BNC. SWISH-E is a fast, powerful, flexible, free, and easy to use system for indexing collections of Web pages or other files. ExRetriever has been designed to be easily ported to other lexical knowledge bases and corpora, including the possibility to query search engines such as Google.

3 ExRetriever

Although this approach seems to be very promising, it remains unclear which is the best strategy for building sense queries from a large-scale knowledge base

[3] `http://www.lsi.upc.es/~nlp/meaning`

[4] `http://swish-e.org`

like WordNet. ExRetriever will explore the trade-off between coverage (collecting large quantities of sense examples) and accuracy (making queries more precise and restrictive, and obviously less productive).

First experiments have been performed using large scale corpora stored locally. This allowed to perform controlled tests and comparisons between different query buiding strategies very fast in order to obtain a more clear view of the knowledge to be used (e.g. regarding PoS, monosemous relatives only, synonyms, direct hypernyms, direct hyponyms, INVOLVED relations, etc.) the query construction (e.g. including or not AND-NOTs with characterizations of the other sense queries), the complete query process (e.g. union set of queries, incremental construction, etc.), the post processing (e.g. using PoS, syntactic or domain filtering), the other languages involved in the project (using the MCR) and corpus.

This tool characterises each sense of a word as a specific query. This is automatically done by using a particular query construction strategy, which is defined *a priory* by an expert. Each different strategy can take into account the information related to words and available into a lexical knowledge base in order to automatically generate the set of queries.

The current version of ExRetriever is able to use different lexical databases through the MCR of MEANING [4] and different corpora (SemCor, BNC, the Web, etc.) through a common API.

In order to easily implement different query construction strategies, ExRetriever has been powered with a declarative language. This language allows the manual definition of complex query construction strategies and it is briefly described in the fowolling section.

4 The Query Language

ExRetriever query language consist on the following three component types: logical operators, functions and constants.

- **Operators** are the usual boolean operators **and** , **or** and **not** .
- **Functions** Currently,
  - **Glos** used to obtain the words appearing in the gloss.
  - **rel** used to obtain the different relations in the lexical knowledge base
  - **nrel** similar to *rel*, but stablishing the maximum polysemy of the returned senses.
- **Constants** can be divided in:
  - **noempty** a parameter for the **Glos** function, used to remove all stopwords from a gloss.
  - **senses** particular senses (e.g. church#n#2)
  - **relations** particular MCR relationships used as parameters to "rel" and "nrel" (e.g. *hypo*).

4.1 Example for chair

In this section we explain, using an example, the construction of a query accordingly to a particular query construction strategy. We apply the query strategy **Meaning1**, { Glos(or,and,noempty) **or** or(nrel(1,syns)) **or** or(nrel(1,hypo))} to the third sense of *chair*. Table 1 provides a brief description of word *chair* in Wn1.6.

The first function *Glos(or,and,noempty)* returns a logical formula which is the target word (i.e. *chair*) and the union set with *or* of the non *noempty* words of the *gloss* of chair#n#3: (*chair* AND (*officer* **or** *presides* **or** *meetings* **or** *organization*)). The second function, *or(nrel(1,syns))* returns the union set with *or* of the monosemous synonyms of chair#n#3: (*chairman* **or** *chairwoman* **or** *chairperson*). Finally, *or(nrel(1,hypo))* returns the union set of the monosemous hyponyms of chair#n#3: **or** (*vice chairman*). Table 2 shows the resulting queries for all the sense of the word *chair* (noun).

sense	**gloss**	**hypo**	**syn**
n#1	*a* seat *for one* person , *with a* support *for the* back	*armchair (2)* barber_chair ...	
n#2	*the* position *of* professor		professorship
n#3	*the* officer *who* presides *at the* meetings *of an* organization	vice_chairman	*president (6)* chairman chairwoman chairperson
n#4	*an* instrument *of* death *by* electrocution *that* resembles *a* chair		electric_chair death_chair hot_seat

Table 1. Sense of *chair* noun in wordNet 1.6

4.2 Examples obtained from SemCor

Once a query strategy is applied to a particular word, we can use the resulting queries in a search engine to retrieve examples for a selected sense. The examples retrieved are structured using XML and include information about their source, the target word and the base sense from which the query is build.

<Example Sentences=”1” src=”brown2/tagfiles/br-l15#104577” > *It contained a desk, files, a typewriter on a stand, and two big leather* <MEANING origPOS=”n” rel=”hypo” synsetSense=”1” synsetLema=”armchair” synsetPOS=”n” baseSense=”1” baseLema=”chair” basePOS=”n” origSense=”1” > *armchairs* </MEANING>.</Example>

chair#n#1:
(*chair* **and** (*seat* **or** *person* **or** *support* **or** *back*))
or
(*barber chair* **or** *chaise longue* **or** *folding chair* **or** *highchair* **or** *feeding chair* **or** *ladder-back chair* **or** *lawn chair* **or** *garden chair* **or** *rocking chair* **or** *straight chair* **or** *side chair* **or** *swivel chair* **or** *tablet-armed chair* **or** *wheelchair*)
char#n#2:
(*chair* **and** (*position* **or** *professor*))
or
(*professorship*)
chair#n#3:
(*chair* **and** (*officer* **or** *presides* **or** *meetings* **or** *organization*))
or
(*chairman* **or** *chairwoman* **or** *chairperson*)
or
(*vice chairman*)
chair#n#4:
(*chair* **and** (*instrument* **or** *death* **or** *electrocution* **or** *resembles*))
or
(*electric chair* **or** *death chair* **or** *hot seat*)

Table 2. Queries for *chair* noun usin **Meaning1**

It is likely that some of the conditions imposed by the query strategies are not available on the corpus, specially when retrieving examples from Internet (e.g. lemma, syntactic functions) or that the original information is not compatible with the knowledge in Mcr (e.g. different MultiWords Expression criteria or different PoS tagset). These issues can affect greatly the performance of the system.

Although Semcor is completely lemmatized (except brownv), in order to simulate the performance in a untagged corpus, we use a special MultiWord Expression module[5] for tagging, lemmatizing and recognizing wordnet multiwords.

Instead of processing the whole corpus, the examples retrieved based on word forms are post-processed, filtering out the incompatible PoS tags or lemmas. Thus, this technique could also be applied to corpora acquired from Internet.

5 Experiments

Within the framework of the Meaning project we designed a preliminar set of tests to validate ExRetriever. Both direct and indirect evaluation experiments of the ExRetriever performance have been designed. In this paper we present the results of the direct evaluation on SemCor.

Using ExRetriver on SemCor we can perform detailed micro-analisys on the data available (preliminar results in [6]). That is, we can easily perform many adjustements for building queries and filtering appropriately those unwanted examples, balancing the trade-off between coverage (we want to obtain all the examples of a particular sense occurring in a corpus) and precision (we want only those corresponding to the particular sense).

Each one of such experiments consists of applying a particular query construction strategy to a set of 73 English words from Senseval-2 *lexical sample* task. The resulting specific queries (one for each sense word) automatically generated by applying each strategy have been tested against Semcor. Due to the small size of Semcor (around 250 thousand words), specific queries are likely to produce poor recall. However, Semcor is the unique sense tagged resource providing large quantities of examples for all-words.

Six different query construction strategies have been tested, some of them inspired in those used by other authors. They are briefly described as follows:

1. **Lea1**: or(nrel(1,syns)) **or** or(nrel(1,hypo)) **or** or(nrel(1,hype))
Inspired in the work presented in [1], this strategy generates a specific query for each word sense by collecting only monosemous relatives (i.e., synonyms, immediate hyponyms and inmediate hypernyms of the sense).
2. **Moldo1**: or(nrel(1,syns))
Used as in [2], this strategy builds each specific query as the set of monosemous synonyms of the particular word sense. In fact, this is a particular case of the previous strategy.
3. **Moldo2**: or(rel(glos))
This method builds a query corresponding exactly to the gloss of the synset.
4. **Moldo3**: Glos(or,and,noempty)
This strategy is a simplified version of the fourth method described in [2]. As we do not parse the glosses, we can not use their head phrases. Instead we only remove the stopwords.
5. **Meaning1**: Glos(or,and,noempty) **or** or(nrel(1,syns)) **or** or(nrel(1,hypo))
In order to increase the coverage of the previous strategies, we added to the previous method, the posibility to query also for their monosemous relatives (synonyms and hyponyms).
6. **Meaning2:**: Glos(or,and,noempty) **or** Glos(or,and,or,rel(hypo),noempty)
The second function of this method builds the query using all the hyponym glosses (removing the stopwords) and their defining senses.

6 Results

Table 3 shows the overall figures for each query when applied to the total 73 words of the test set. *Ok* stands for correctly detected examples of the respective senses of the word. Those incorrectly assigned senses are labeled with *Ko*. *NoTag* corresponds to non sense annotated word occurrences occurring in Semcor (those coming from bronv files). *#Sense* stands for the total number of sense

Q	Ok	Ko	NoTag	#Sense	P	R	F1	WSC
Lea1	851	10	371	23254	98,84	3,66	7,06	23
Moldo1	153	1	83	3241	99,35	4,72	9,01	10
Moldo3	1987	22474	1303	7611	8,12	26,11	12,39	47
Meaning1	2314	22617	1415	9490	9,28	24,38	13,44	54
Meaning2	4513	37688	2986	17171	10,69	26,28	15,20	58

Table 3. Overall figures

occurrences occurring in Semcor (i.e. the total coverage). As each query asks for different relatives, they also obtain different number of possible sense occurrences. Finally, *P*, *R* and *F1*, correspond to precision, recall and F–measure, respectively. Unfortunately, these measures do not show if the examples retrieved cover all the senses of a word. This is a crucial issue if we want to use the acquired examples to train supervised WSD systems. Besides the frequency of a word sense, the way in which the queries are built and the knowledge contained in Mcr biases the retrieved sense examples. Thus, we have defined WSC to calculate the Word Sense Coverage of the examples retrieved from the corpus.

$$WSC = 100 \times \sum_{w=1}^{n} \frac{SensesWithinRetrievedExamples(w)}{SensesWithinCorpus(w)}$$

When applying systematically the same method to all the words, **Moldo1** and **Lea1** strategies obtain the best precision (around 99%). However, **Meaning1**, **Meaning2**, **Moldo3** methods obtain much better recall (about 25% vs 5%). **Meaning2**, the best WSC obtaining examples for 58% of the senses. **Moldo2** strategy do not provide results in SemCor, as this method is looking for the complete synset gloss. Obviously, in a small corpus such as Semcor this is highly improvable.

In summary, the results in table 3 show the trade-off between precision, coverage and Word Sense Coverage. **Lea1** and **Moldo1** has high precision but poor coverage and WSC. While **Moldo3**, **Meaning1** and **Meaning1** has more recall and WSC but less precision.

7 Conclusions and future work

In this paper, ExRetriever, a query-based system to extract sense examples from corpus has been described. Some preliminar experiments have been presented. They have been used to evaluate the performance of different types of query construction strategies. Using ExRetriever, new strategies can be easily defined, executed and evaluated.

We plan to experiment other strategies. For instance, performing full parsing on the glosses could help discarding irrelevant words from glosses. In addition, using the knowledge already contained into the Mcr (e.g., selectional preferences acquired from the BNC, eXtended WordNet, domain information, the Topic Signatures acquired from the Web, etc.) could be useful knowledge to better model sense words as queries. Moreover, we plan to use alternative schemata for building queries, such as the incremental process performed by [1].

Another promising line of research will follow [7]. This work presents a theoretically motivated method for removing unwanted meanings directly from the original query in vector models. Irrelevance in vector spaces is modelled using orthogonality. Using this approach, query vector negation removes not only unwanted strings but unwanted meanings. This method is applied to standard IR systems, processing queries such as “play NOT game”. This work presents an algebra to operate with word vectors rather than words. It seems, following this approach, that most of the errors produced because of the substitution of the target word for their relatives can be avoided.

We also plan to perform indirect evaluations using supervised WSD systems on the acquired sense examples. Once acquired a sense tagged corpus using ExRetriever, we will use several Machine Learning algorithms to perform several cross-comparisons with respect to other sense tagged resources (SemCor, DSO and those resources provided by Senseval).

8 Acknowlegments

This work is suported by the European Comision (MEANING IST-2001-34460). Our research group, TALP Research Center, is recognized as a Quality Research Group (2001 SGR 00254) by DURSI, the Research Department of the Catalan Government.

References

1. Leacock, C., Chodorow, M., Miller, G.: Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics **24** (1998) 147–166
2. Mihalcea, R., Moldovan, I.: An Automatic Method for Generating Sense Tagged Corpora. In: Proceedings of the 16th National Conference on Artificial Intelligence, AAAI Press (1999)
3. Agirre, E., Martinez, D.: Exploring Automatic Word Sense Disambiguation With Decision Lists and the Web. In: Proceedings of the COLING workshop on Semantic Annotation and Intelligent Annotation, Luxembourg (2000)
4. Atserias, J., Villarejo, L., Rigau, G., Agirre, E., Carroll, J., Magnini, B., Vossen, P.: The Meaning Multilingual Central Repository. In: Second International WordNet Conference-GWC 2004, Brno, Czech Republic (2004) 23–30 ISBN 80-210-3302-9.
5. Arranz, V., Atserias, J., Castillo, M.: Multiword Expressions for Word Sense Disambiguation. Technical report, of the LSI Department. LSI-04-47-R. Universitat Politecnica de Catalunya (2004)
6. Fernández, J., Castillo, M., Rigau, G., Atserias, J., Turmo, J.: Automatic Acquisition of Sense Examples using ExRetriever. In: LREC’04. (2004) 25–28
7. Widdows, D.: Orthogonal Negation in Vector Spaces for Modelling Word-Meanings and Document Retrieval. In: Proceedings of 41th annual meeting of the Association for Computational Linguistics (ACL’2003), Sapporo, Japan (2003)