Accurate Semantic Annotations via Pattern Matching

Adrian Novischi
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75083-O688 U.S.A.
adrian.novischi@student.utdallas.edu

Abstract

This paper addresses the problem of performing accurate semantic annotations in a large corpus. The task of creating a sense tagged corpus is different from the word sense disambiguation problem in that the semantic annotations have to be highly accurate, even if the price to be paid is lower coverage. While the state-of-the-art in word sense disambiguation does not exceed 70% precision, we want to find the means to perform semantic annotations with an accuracy close to 100%. We deal with this problem in the process of disambiguating the definitions in the WordNet dictionary. We propose in this paper a method that is able to tag words with high precision, using pattern extraction followed by pattern matching. This algorithm exploits the idiosyncratic nature of the corpus to be tagged, and achieves a precision of 99% with a coverage of 6%, measured on a WordNet subset, respectively more than 12.5% coverage estimated for the entire WordNet.

Motivation

In the Natural Language Processing community, WordNet (**?**) is well known as a valuable resource: more and more applications that require machine readable dictionaries or world knowledge encoded in semantic networks use WordNet. There are two sides to be exploited in WordNet. First, the ontological side proved to be useful in Information Retrieval applications, as well as conceptual and semantic indexing, where the ISA relations defined across concepts are enough to improve retrieval effectiveness. The other side, where WordNet is seen as a lexicon rather than an ontology, was employed in many applications including word sense disambiguation, logic forms and answer proving (**?**), topical relations (**?**), and others. This latest type of applications do require a large number of concepts and semantic relations among concepts. In addition to the relations that are explicitly encoded in WordNet, a multitude of new implicit relations can be derived via gloss definitions and examples. To enable this derivation process, supplementary information is needed, and the knowledge about the semantic meaning of the definition words is a must. This was the main reason that triggered our interest in creating an extended version of WordNet where all words in the glosses are tagged with their appropriate senses.

Things will hopefully become clearer through an example, which will also make apparent our motivation in doing this work. Consider for instance the problem of deriving topical relations, already shown to be valuable information for tasks such as texts coercion and cohesion (**?**). Starting with an input word, we would like to derive a *cloud* of concepts semantically related to that word.

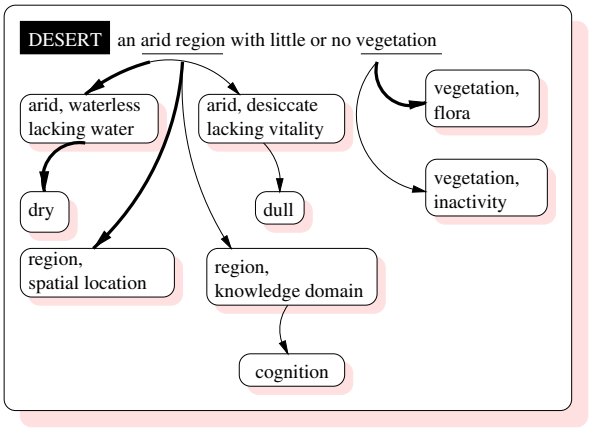

Figure 1: Semantic relations across WordNet concepts

Figure 1 shows an example of relations that can be derived across concepts in WordNet. Starting with the concept *desert*, one might want to derive a list of related concepts, such as *dry*, *sun*, *cactus*, etc. Lists of related words are certainly useful for a large range of applications, including the relevance feedback methodology in Information Retrieval, Word Sense Disambiguation, and others. Deriving such a list is a doable task if we make use of the large number of connections originating from a word definition. The problem we now face regards the validity of these newly introduced connections. For instance, as shown in Figure 1, there is a

multitude of concepts that may be easily derived by simply following gloss connections. Unfortunately, not all such connections are valid. We do not want to add *knowledge domain* or *dull* to the context of *desert*, instead *arid* and *dry* constitute informative concepts for this domain. Out of all possible connections that can be set via gloss relations we want to keep only those that are represented in bold. Having the capacity to make the appropriate choice in selecting valid links, implies in the first place knowledge about the meaning of definition words. If the senses of *arid*, *region* and *vegetation* are known in the input gloss, we expect to have only the bold relations set across concepts, and avoid incorrect word associations.

TERMINOLOGY Synset S = the basic unit in WordNet = SYNonym SET = set of synonym words Gloss G = the definition associated with a synset Word W = the word disambiguated by a procedure
Procedure 1. Monosemous words. *ID*: MONOS Identify words W with only one sense in WordNet
Procedure 2. Same hierarchy relation. *ID*: HYPER Identify words W belonging to the same hierarchy as the synset S of the gloss.
Procedure 3. Lexical parallelism relation. *ID*: LEXPAR Identify words W involved in a lexical parallelism (words connected with conjunctions or comma). These words should belong to the same hierarchy, and therefore they can be disambiguated.
Procedure 4. SemCor bigrams. *ID*: SEMCOR Get contextual clues regarding the usage of a word form W, based on SemCor (a corpus tagged with WordNet senses). For now, we only rely on SemCor bigrams.
Procedure 5. Cross reference. *ID*: CROSSR For a word W ambiguous in the gloss G belonging to the synset S, find if there is a reference from one of the senses of the word W to the words in the synset S.
Procedure 6. Reversed cross reference. *ID*: RCROSSR For a word W ambiguous in the gloss G, find if there is a reference from G to a word in one of the synsets of W.
Procedure 7. Distance between glosses. *ID*: GDIST Given a word W ambiguous in the gloss G, find the number of common words between the glosses attached to each of W senses and the gloss G.

Table 1: Best performing methods for the semantic annotation of WordNet glosses

We have to solve therefore the task of annotating words in glosses with their appropriate sense in WordNet. WordNet 1.7 contains almost 110,000 glosses, adding up to about 3,5 million words. From these, more than 500,000 are open class words and will be undergoing the disambiguation process, which is far by being a doable task without means for automatization.

The difficulty of the task turns out to be tremendously higher than simple word sense disambiguation, as the conditions posed are for very high accuracy. The state-of-the-art in word sense disambiguation with respect to WordNet is slightly under 70% (see the Senseval web page http://www.sle.sharp.co.uk/senseval2/), which is far from the constraint we have in this project of close to 100% accuracy.

Several methods have been previously proposed for the semantic disambiguation of gloss words (**?**). Table 1 summarizes the best performing methods implemented so far. They attain an overall coverage of 60% with a precision of over 95%. We introduce in this paper an additional method that exploits the idiosyncrasy of the corpus formed with the entire set of definitions. The algorithm consists in searching patterns in the corpus, manually disambiguate them and then perform pattern matching for improved coverage and high precision.

It was noticed that glosses have certain characteristics that could eventually help towards disambiguation. With the method we propose here, we want to first find these specific features and then exploit them during the process of automatically solving semantic ambiguity.

To better explain the type of features we are looking for, let us take a closer look at some examples of WordNet definitions. Table 2 lists eight glosses drawn from WordNet 1.7 databases. It turns out that many lexical patterns are used in the definitions, and learning what these patterns are will result in a tool for performing disambiguation. For instance, if we once disambiguate the pattern *to a ... degree*, there is no need to repeat the same process a second time. Whenever we encounter this pattern in a new ambiguous gloss, we can assign the same sense as it was done for the first time. This pattern occurs 90 times in the WordNet database. The other three patterns observed in Table 2 have even higher frequencies: *in a ... manner* (1160), *the act of* (1138), *of or relating to* (1674).

Word	Definition
highly, extremely	to a high degree or extent
widely	to a great degree
singularly	in a singular manner or to a singular degree
possessively	in a possessive manner
arrival	the act of arriving at a certain place
incursion	the act of entering some territory
egoistic	of or relating to the self
graduate,	of or relating to studies

Table 2: Examples of WordNet glosses

Semantic annotations based on patterns

There are two distinct phases in the process of performing semantic annotations using patterns. First, we need to extract valid patterns from the target corpus.

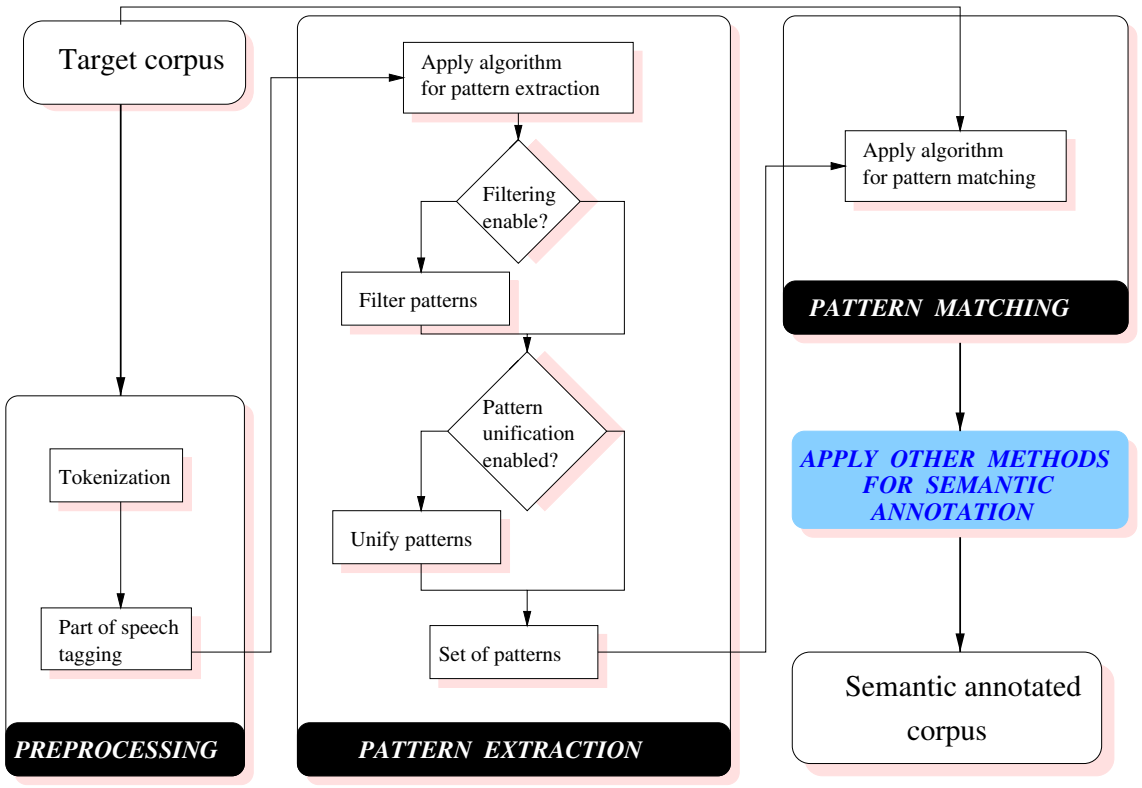

Figure 2: Semantic annotations using pattern matching

The issue that we have to address here is what type of patterns to look for and how to filter our meaningless patterns. During this stage we also face the problem of *unification*: overlapping patterns have to be unified to their most general format.

The second stage deals with pattern matching. The set of patterns extracted during the first step, after being semantically disambiguated, are applied on the original corpus. As stated earlier, the pattern based annotation process aims at exploiting the idiosyncratic nature of a corpus, and therefore both pattern extraction and pattern matching are performed on the same target corpus. In our case, the corpus is formed with all dictionary definitions found in WordNet.

Figure 2 shows the main steps performed in pattern extraction and pattern matching. We start with a raw corpus and end up with semantically annotated text. Even though we use this procedure to solve the semantic ambiguity of words in WordNet definitions, the algorithm is applicable to other types of corpora.

Three main steps are identified in this figure: (1) preprocessing; (2) pattern extraction and (3) pattern matching. We detail below each of these steps and present the algorithms for pattern extraction and pattern matching.

Step 1: Preprocessing

This is just a preparation step, which merely changes the corpus into a format suitable for the stages that follow. First, definitions are extracted from the WordNet databases. This results in four separate files, one for each open class part of speech. Subsequently, the text is tokenized and part of speech tagged (**?**).

Step 2: Pattern extraction

The decision of what rules to employ during pattern extraction has strong impact on the *quality* and *quantity* of extracted patterns. More patterns we extract in this step, the higher coverage they will ensure in the matching phase. Therefore, *quantity* directly affects the *recall* of our method. On the other hand, wrong patterns will result in wrong semantic annotations. Hence, patterns *quality* is the principal factor influencing *precision*.

The following rules currently constitute our pattern extraction guidelines.

Rule 1. Extract all N successive words, with $N \in [2-5]$. The rationale behind this rule is the fact that in many cases strings of consecutive words place a constraint over the possible word meanings. For instance, the verb *to relate* will most probably have the same sense in all occurrences of the string *of or relating to*. We refer to the type of patterns extracted with this first rule as *N-patterns*.

Rule 2. Extract all sequences of words that follow the pattern *"[N words] ... [M words]"*. The following constraints have to be satisfied for a pattern to be extracted:
∘ $N \in [1-5] \vee M \in [1-5]$.
∘ The distance between the two sides, represented with "..." includes no more than T tokens, with T=7.
∘ At least one open class word is included in each such pattern. The purpose of this condition is to avoid patterns such as *to the*, *of a*, etc., which bring no useful information for our task.

With these rules, we extract all possible patterns from the entire corpus formed with WordNet definitions. Next, the patterns are sorted based on their frequency and we keep only those with a number of occurrences larger than a given threshold. Different threshold values were considered during our experiments, and we show in the following section the figures attained for each such value.

To determine the values for the M and N parameters, we count all tokens found in a string, except those marked with the tags in a *stop-tag* list. Currently, the *stop-tag* list consists of all punctuation signs; additionally, the following parts of speech are not considered: *CC, DT, FW, LS, SYM, UH*[1]. This means that for instance the string *and/CC today/NN* has associated a value of N=1, because we do not count *and/CC* towards the final value of N. This additional constraint is intended to avoid the extraction of meaningless patterns.

Moreover, two other optional procedures can be activated during the pattern extraction stage, as shown in Figure 2.

Filtering can be applied to filter out meaningless patterns. For instance, we have observed that *M-N patterns* including an adjective as the only open class word are usually not useful. We have determined so far twelve general filters that can be applied to avoid inutile patterns:

1. M-N patterns containing only adjectives.
2. M-N patterns containing only adverbs.
3. All patterns with an un-closed parenthesis.
4. M-N patterns containing only the word "that" at one side.
5. N successive words patterns that ends with "that".
6. M-N patterns containing only a conjunction at one side.
7. M-N patterns containing only the preposition "to" at the right side.
8. M-N patterns containing only the preposition "of" at one side.
9. M-N patterns containing only an article ("/DT") at the right side.
10. N successive words patterns of the form "to [verb]".
11. M-N patterns with a punctuation at one side.
12. M-N patterns that have only a modal verb or "to be" verb at the right side ("can", "is", "are").

Unification is a step designated to combine together similar patterns to their most general format. For instance, the following patterns:
relating ... characteristic,
relating to ... or characteristic,
relating ... characteristic of,
relating to ... characteristic of,
relating to ... or characteristic of,
can be all unified to one general pattern *relating ... characteristic*. There is a tradeoff between unification and pattern disambiguation. Sometimes the open class words of the most general pattern does not have the same sense in all occurences of the pattern. Therefore in this case we should consider a more specific pattern.

Step 3: Pattern matching

The following algorithm is applied to perform pattern matching.

---

1. Read in a gloss.
2. For each word W ambiguous in the gloss:
  2.1. Find if there is any pattern containing word W.
  2.2. From all patterns containing the word W, select only those that are applicable on the current gloss.
  2.3. If there is more than one pattern applicable:
    2.3.1. Give priority to N-patterns.
    2.3.2. Give priority to longest patterns.
  2.4. Assign to W the sense found in the pattern.

Example. Consider for instance the definition of the verb *come_close*, which is *"nearly do something"*. The verb *do* in this gloss is retrieved in the pattern *do/VB/2 something/NN/1* and therefore we solve its semantic ambiguity based on this match and assign a sense to both words *do* and *something*.

Application on data sets

We have evaluated our pattern extraction and matching algorithms on two sets of glosses and on the entire Wordnet. The first set of glosses came from one of the WordNet hierarchies, namely *verb.social*, with 1057 verb synsets. The second set of glosses came from *noun.artifact* hierarchy with 3000 noun synsets. Wordnet contains almost 110,000 synsets.

In step 1, all glosses from each data set are tokenized and part of speech tagged. Next, we extract patterns following the two rules described in the previous section. We decided that the patterns threshold for number of occurences to be three i.e. to keep the patterns which occur at least three times in a data set. This process

[1]They correspond to conjunctions, determiners, interjections, symbols. See Treebank tagging for details regarding part of speech notations.

results in 357 patterns occurring in the *verb.social* hierarchy, 6461 patterns in *noun.artifact* hierarchy and 531,245 patterns on Wordnet. These patterns were sorted in a descendant order of their frequencies.

We applied unification and filtering to the extracted patterns and the number of patterns that must be analyzed and disambiguated was significantly reduced. From the remaining patterns we have manually disambiguated 68 patterns from *verb.social* hierarchy, 145 patterns from *noun.artifact* hierarchy, and 131 patterns from Wordnet. Table 3 lists the top ten extracted patterns from Wordnet.

Pattern	Freq.
relating/VBG/2 to/TO	2415
genus/NN/2 of/IN	1925
in/IN ... manner/NN/1	1774
the/DT act/NN/2 of/IN	1137
used/VBN/1 in/IN	1103
consisting/VBG/2 of/IN	1101
used/VBN/1 to/TO	1045
having/VBG ... flowers/NNS/2	782
used/VBN/1 as/IN	752
of/IN the/DT genus/NN/2	752

Table 3: Top ten patterns extracted from Wordnet

These manually disambiguated patterns were applied back to the corresponding data sets and and we got 6% coverage for *verb.social* hierarchy, 8.6% coverage for *noun.artifact* hierarchy and 6.6% on Wordnet. We have evaluated the disambiguation accuracy for all the words from *verb.social* and *nount.artifact* hierarchy and for 1000 ramdom disambiguated words from Wordnet. A precision of 98% or 99% was observed in each case. We consider this result to be encouraging. Our task is to perform extremely accurate semantic annotation of gloss words, and therefore every single word that is disambiguated with high precision is highly praised. Table 4 presents various results for the pattern extraction and pattern matching algoritms.

Conclusions and future work

This paper relates to a project that is in progress. The task of automatically labeling gloss words with semantic tags is very difficult, given the high accuracy required in this project.

We have presented an algorithm that does automatic pattern extraction and pattern matching with the purpose of performing semantic annotations. This algorithm exploits the idiosyncratic nature of the corpus to be tagged, and succeeds in identifying repetitive expressions encountered in text. Experiments performed on a two data sets and the entire WordNet resulted in semantic annotations that cover 6.6% of the words with a precision of 98%, proving the validity of the approach.

	verb.social	noun.artifact	Wordnet
Number of glosses	1057	3000	109,377
Number of extracted patterns	357	6461	531,245
After unification	233	2908	134,183
After unification and filtering	150	1285	74,867
Number of disambiguated patterns	68	145	131
Number of words that can be disambiguated	3979	18831	560871
Number of words disambiguated	242	1635	37243
Coverage	6%	8.6%	6.6%
Precision	99%	98%	98%

Table 4: Pattern matching and pattern extraction

Acknowledgement

This work was supported by the NSF Grant EIA-0078854. The author is indebted to Rada Mihalcea and Dan Moldovan for their collaboration and guidance on this work.

References

Brill, E. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics* 21(4):543–566.

Harabagiu, S., and Moldovan, D. 1998. *Knowledge Processing on an Extended WordNet.* The MIT Press. 289–405.

Mihalcea, R., and Moldovan, D. 2001. eXtended WordNet: progress report. In *NAACL 2001 Workshop on WordNet and Other Lexical Resources: applications, extensions and customizations*, 95–100.

Miller, G. 1995. Wordnet: A lexical database. *Communication of the ACM* 38(11):39–41.

Moldovan, D., and Rus, V. 2001. Logic form transformations of WordNet and its applicability to Question Answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Lingusitics (ACL-2001)*, 394–401.