# A Gloss-centered Algorithm for Disambiguation

**Ganesh Ramakrishnan**
Dept. of C.S.E
IIT Bombay
India - 400076

**B. Prithviraj**
Dept. of C.S.E
IIT Bombay
India - 400076

**Pushpak Bhattacharyya**
Dept. of C.S.E
IIT Bombay
India - 400076

{hare,prithvir,pb}@cse.iitb.ac.in

## Abstract

The task of word sense disambiguation is to assign a sense label to a word in a passage. We report our algorithms and experiments for the two tasks that we participated in *viz.* the task of WSD of WordNet glosses and the task of WSD of English lexical sample. For both the tasks, we explore a method of sense disambiguation through a process of "comparing" the current context for a word against a repository of *contextual clues* or *glosses* for each sense of each word. We compile these glosses in two different ways for the two tasks. For the first task, these *glosses* are all compiled using WordNet and are of various types *viz.* *hypernymy* glosses, *holonymy mixture*, *descriptive glosses* and some hybrid mixtures of these glosses. The "comparison" could be done in a variety of ways that could include/exclude *stemming*, expansion of one gloss type with another gloss type, *etc*. The results show that the system does best when stemming is used and glosses are expanded. However, it appears that the evidence for word-senses ,accumulated through WordNet, in the form of glosses, are quite sparse. Generating dense glosses for all WordNet senses requires a massive sense tagged corpus - which is currently unavailable. Hence, as part of the *English lexical sample task*, we try the same approach on densely populated glosses accumulated from the training data for this task.

## 1 Introduction

The main idea behind our approach for both the WSD tasks is to use the context of a word along with the *gloss* or *description* of each of its senses to find its correct sense. The similarity between the context and each sense of the word is measured and the word-sense with the highest similarity measure is picked as most appropriate, that with second highest similarity is ranked second and so on.

Glosses have been used by authors in the past for WSD (Lesk, 1986). The novelty in our approach, for the task of disambiguation of extended WordNet is in the way we generate our descriptions or glosses. Also, an additional novelty in the second task, is in our use of textual proximity between words in the neighborhood of the word to be disambiguated and the words in the glosses of each of its senses.

## 2 Glosses

### 2.1 Descriptive glosses

A word, with its associated part of speech and an associated sense number, has a description.

**Description for fifth noun sense of "tape"** memory device consisting of a long thin plastic strip coated with iron oxide; used to record audio or video signals or to store computer information

Figure 1: An example *descriptive-gloss* for "tape" from WordNet

We call these descriptions - *descriptive glosses*. For word-senses picked up from WordNet, the WordNet glosses are the descriptive glosses. WordNet glosses also contain example usages of the word. We have excluded the examples from descriptive glosses

For other word-senses, the descriptions could come from glossaries (like glossaries of software terms), encyclopedias (for names of people, places, events, pacts *etc*), world fact books, abbreviation lists *etc*. Examples glosses picked up from above sources are listed below.

**descriptive-gloss for "piccolo"** an instrument of the woodwind family. Most of these instruments were once of made of wood, and because they are played by blowing with air or wind, they are called woodwind.

Figure 2: Examples of descriptive glosses for non-WordNet words picked from glossaries

### 2.2 Hypernymy glosses

The gloss for a particular sense of a word could also describe what hierarchical categories it belongs to.

For instance, the hierarchical categorization of the $1^{st}$ noun sense of the word "Vesuvius" is:

```
Vesuvius#n#1
=> volcano
 => mountain, mount
  => natural elevation, elevation
   => geological formation
    => natural object
     => object, physical object
      => entity
```

Based on this hierarchical categorization of the first noun sense of "Vesuvius", we describe its *hypernymy-gloss* as the collection of all nodes in its hypernymy-path to the root - *viz.* "entity".

**Hypernymy gloss for first noun sense of "Vesuvius"** {volcano}, {mountain, mount}, {natural elevation, elevation}, {geological formation, formation}, {natural object}, {object, physical object}, {entity}

Figure 3: Hypernymy-based gloss for the first sense of Vesuvius(noun)

Whereas *descriptive-glosses* can be derived even for word-senses not present in WordNet, *hypernymy-glosses* require classification of word-senses into nodes into an ontological structure - like the hypernymy hierarchy of WordNet. This is not that easy to procure for words not present in WordNet.

### 2.3 Hyper-Desc($n$) glosses

This category of gloss was developed for each word-sense by concatenating the descriptive glosses of a word-sense with the glosses of its hypernyms, all the way upto height $n$. Hyper-Desc($*$) glosses denotes concatenating descriptive glosses all the way upto the root.

### 2.4 Holo-Desc($n$) glosses

The specification of these glosses is same as of Hyper-Desc($n$) glosses, except that holonyms are considered here instead of hypernyms.

#### Handling Named Entities

One possible solution, and the one that we actually resort to, is to find the named entity tag for a token (if one exists) and then map the tag to a node in WordNet. For example, the token "President Musharraf" is not present in WordNet. But this token can be tagged as a *PERSON* and *PERSON* could be mapped to a node in WordNet - *viz.* the first noun sense of "person" (person#n#1). Similarly, the token "$26^{th}$ December 2003" has a DATE named-entity tag. DATE could be translated to the $7^{th}$ sense of the word "date" (date#n#7) in WordNet.

Thus, the *glosses* of named entites, which dont find their entries into WordNet could be evolved from their named-entity tags. This information is valuable for disambiguating the surrounding words. For the seneval task, we have built our own Named Entity tagger that uses gazetteers and context-sensitive grammar rules for tagging named entities.

#### Context of a word

The context of the word to be disambiguated (target word) can be evolved in several possible ways.

1. The passage in which the target word lies can be tokenized and the set of tokens are considered the context for that word.

2. In addition to tokenizing the passage as described above, each token is also subjected to *stemming* using the porter stemming algorithm (Porter, 1980). The corresponding set of stemmed tokens form the context. This option is abbreviated as ST in table **??**.

3. The passage can be part of speech tagged. In the case of SemCor and Extended WordNet, the part of speech tags have already been assigned manually. In the absence of a manual POS tags, we use the QTag part of speech tagger (Manson, 1980). And each part of speech tagged word is expanded to the concatenation of the glosses of all its word-senses. The collection of all tokens in the expansions of all words in the passage put together forms the context for the target word. In table **??**, this option is abbreviated as FG.

## 3 Similarity metrics

Another parameter for measuring the similarity between the *context of a word* and the gloss of each of its senses is the similarity metric.

The similarity between two sets of tokens is found by constructing vectors of counts from the two vectors and finding similarity between the vectors.

### 3.1 Cosine similarity

One standard metric of similarity, as used in information retrieval, is the cosine-similarity. We find

the cosine similarity between the *term frequency-inverse gloss frequency (tfigf)* vectors of the two sets. The *inverse gloss frequency (igf)* of a token is the inverse of the number of glosses which contain that token and it captures the "commonness" of that particular token.

There have been fancier definitions of similarity in literature (Lin, 1998) which involve information theoretic measures of similarity between word-senses, based on the hypernymy path and DAG structure of WordNet. These methods are heavily dependent on frequencies of synsets in a sense-tagged corpus. The idea is that two word-senses are highly related if their subsuming synsets are highly information bearing - or in other words, have high information content. Information content is computed from a sense tagged corpus - word-senses with *high* frequencies of occurrence have *low* information content. This brings in the the problem of data sparsity - because sense-tagged corpora are very scarce and of short size. Their coverage of synsets is poor as well. Hence there is the danger of making the similarity measure biased toward the sense-tagged corpus.

Also, these methods are very slow and CPU intensive, since finding similarity between two word-senses at run time involves traversing the WordNet graph, in the direction of hypernymy links, up to the least common ancestor.

On the other hand, a cosine similarity on tfigf vectors built from *hypernymy-glosses*, gives a low similarity value between word-senses whose hypernymy-glosses overlap in very frequently occurring synsets relative to the synsets which are not common to their glosses. This is because *igf* implicitly captures the information content of a synset - the higher the igf - higher is the information content of a synset. The purpose served by a sense-tagged corpus is cumulatively served by the collection of hypernymy glosses of all the WordNet synsets. This method is also more reliable since the *igf* values come from WordNet which is very exhaustive, unlike sense tagged corpora (like SemCor) which will have bias and data-sparsity in terms of which words occur in the corpus and which sense is picked for a word. (The reader might want to note some work which has been done to illustrate that words can inherently have multiple senses in a given context).

The cosine similarity on tfigf vectors built from *descriptive glosses* is very much like the similarity found between document and query vectors, since the tokens in descriptive glosses are regular words. Cosine similarity is intuitively the most useful similarity measure on descriptive glosses since cosine similarity of *tfigf* vectors takes care of stop words and very non-informative words like "the" *etc.*

### 3.2 Jaccard similarity

Another metric of similarity is the *jaccard similarity*. Jaccard similarity between two sets of tokens (glosses) is computed as $\frac{|A \cup B|}{|A \cap B|}$. Here $A$ and $B$ are the two glosses.

Jaccard similarity is appealing only if the glosses used are *hypernymy-glosses*.

### 3.3 Asymmetric measures of similarity

The above two were symmetric measures of similarity. A third asymmetric similarity measure is one that takes a value of 0 if the intersection of the glosses of two word-senses is not equal to the gloss of one of the word-senses. Else, the similarity is equal to one of cosine or jaccard similarity measures. This means that there are actually two asymmetric similarity measures - one due to jaccard and the other due to cosine.

## 4 Main Algorithm

For each word, a set of content words in its surrounding was found and the similarity of this set with with the gloss of each sense of the word was measured. Cosine similarity measure was used for all the experiments. The senses were then ordered in decreasing value of scores. The word-sense with highest similarity measure was picked as its most appropriate sense. Following were the parameters used in the sense-ranking algorithm.

### 4.1 Parameters

1. **GlossType :** The type of gloss being used in the algorithm. It can be any one of the four outlined in section 2.

2. **Similarity measure:** The cosine similarity measure was used in all the experiments.

3. **Stemming :** Sometimes the words in the context are related semantically with the gloss of the ambiguous word but they may not be in the same morphological form. For example, suppose that the context contains the word *Christian* but the gloss of the word contains the word *Christ*. The base form of both the words is *Christ* but since they are not in the same morphological form they will not be treated as common words during intersection. Stemming of words may prove useful in this case, because after stemming both will give the same base form.

4. **FullContextExpansion :** This parameter determines whether or not the words in the context should be expanded to their glosses. This feature expands the context massively. If set true the gloss of each sense of each context word will be included in the context.

5. **Context size :** The context size can be 1 or 2 sentences *etc.* or 1 or 2 paragraphs *etc.*

## 5 Experimental Results

The algorithms were evaluated against Semcor and was also used in Senseval-3 competition. We present results in this section.

### 5.1 Results for Semcor

For preliminary experiments, we chose the Semcor 1.7 corpus. It has been manually tagged using WordNet 1.7 glosses. The baseline algorithm for sense-tagging of Semcor was of picking a sense for a word, as its correct sense, uniformly at random. This gave us a precision measure of 42.5% for nouns and 23.2% for verbs. Tables 2, 3, 4 and 5 report precision for WSD on Semcor, using our algorithm, with different parameter settings. We see that the algorithm certainly makes a difference over the baseline algorithm.

PrRank1 and PrRank2 (precision at rank 1 and 2 respectively) denote the percentage of cases where the highest scoring sense is the correct sense or one of first two highest scoring senses is the correct sense, respectively. Our recall measures were the same as precision because every word was assigned a sense tag. In the event of lack of any evidence for any sense tag, the first WordNet sense(the most frequent sense) was picked.

Also note that acronyms in table 1 have been employed for parameters in the subsequent tables.

| Stemming | ST |
|---|---|
| ContextSize (in number of sentences) | WS |
| FullContextExpansion | FG |
| POS | P |
| PrRank1 (%) | R1 |
| PrRank2 (%) | R2 |

Table 1: List of acronyms used

### 5.2 Results for Senseval-3 task

For the Senseval task, we employed hypernym glosses. The remaining parameters and the results are tabulated in table 6.

We find results quite poor. We performed additional experiments with modified paramater set and find great improvement in numbers. Moreover, we pick the first WordNet sense in event of lack of any evidence for disambiguation. Hence, in the next reported experiment, the recall values are all same as precision. Based on our experience with the SemCor experiments, we used *Hyper-Desc(*2*)* glosses and a context size of 1 sentence. The results are presented in the table 7. The baseline precisions we obtained were by sampling word-senses uniformly at random. The baseline precision was 45.7% for nouns and 25.4% for verbs.

| ST | WS | FG | P | R1 | R2 |
|---|---|---|---|---|---|
| No | 1 | T | n | 50.3 | 69.2 |
| No | 1 | T | v | 29.1 | 50.1 |
| No | 1 | F | n | 71.4 | 83.9 |
| No | 1 | F | v | 41.5 | 64.7 |
| No | 2 | T | n | 47.7 | 66.8 |
| No | 2 | T | v | 26.4 | 44.8 |
| No | 2 | F | n | 49.1 | 67.7 |
| No | 2 | F | v | 24.9 | 41.4 |
| No | 3 | F | n | 47.3 | 66.5 |
| No | 3 | F | v | 25.5 | 41.6 |

Table 2: Results for *Hypernymy* glosses

| ST | WS | FG | P | R1 | R2 |
|---|---|---|---|---|---|
| Yes | 1 | T | n | 62.2 | 80.32 |
| Yes | 1 | T | v | 36.6 | 59.5 |
| No | 2 | T | n | 57.04 | 77.21 |
| No | 2 | T | v | 34.2 | 56 |
| Yes | 2 | T | n | 45.8 | 65.8 |
| Yes | 2 | T | v | 22.8 | 40 |
| Yes | 2 | F | n | 58.13 | 78.04 |
| Yes | 2 | F | v | 34.03 | 56 |
| Yes | 3 | F | n | 54.7 | 76.3 |
| Yes | 3 | F | v | 31.4 | 51 |
| Yes | 3 | T | n | 47.7 | 66.1 |
| Yes | 3 | T | v | 24.4 | 42.5 |

Table 3: Results for *Hyper-Desc(*2*)* glosses

## 6 English Lexical Sample Task

The results of our gloss based disambiguation system show that an optimal configuration of the parameters is essential to get good results. *Hyper-Desc(*2*)* glosses together with stemming seem to almost always give better results than other. But it may be worthwhile finding out the weight-age for different types of glosses and use all of them together. However - the algorithm performs better than the baseline algorithm, it still falls short of a decent precision that is generally a pre-requisite for the use of WSD in Machine Translation - 90%. One obvious reason for this is that no matter how we try

| ST | WS | FG | P | R1 | R2 |
|---|---|---|---|---|---|
| No | 1 | T | n | 43 | 61.5 |
| No | 1 | T | v | 21.4 | 35.8 |
| Yes | 1 | T | n | 41.3 | 59.3 |
| Yes | 1 | T | v | 21.1 | 36 |
| No | 2 | F | n | 53.6 | 74.9 |
| No | 2 | F | v | 29.7 | 50.6 |
| No | 3 | F | n | 50.9 | 73.1 |
| No | 3 | F | v | 29 | 47.8 |

Table 4: Results for *Hyper-Desc(∗)* glosses

| ST | WS | FG | P | R1 | R2 |
|---|---|---|---|---|---|
| No | 1 | T | n | 49.18 | 71.5 |
| No | 1 | T | v | 26.37 | 43.8 |
| No | 2 | F | n | 62.75 | 79.7 |
| No | 2 | F | v | 37.5 | 58.6 |
| No | 2 | T | n | 48.2 | 73.2 |
| No | 2 | T | v | 26 | 43.3 |
| No | 3 | T | n | 48.5 | 74.3 |
| No | 3 | T | v | 25 | 43.5 |
| No | 3 | F | n | 61.08 | 77.75 |
| No | 3 | F | v | 35.6 | 54.7 |

Table 5: Results for *Holo-Desc(∗)* glosses

to use WordNet, the descriptive glosses of WordNet are very sparse and contain very few contextual clues for sense disambiguation. In the task of *English Lexical Sample*, we further develop the algorithm describe for the previous task and use relatively dense glosses from the training set. The large size of the glosses require us to modify the architecture for ranking glosses. We use and inverted index for indexing the glosses and treat the context of the word to be disambiguated as a query. The senses of the word are ranked using the same set of parameters as described for the earlier task.

### 6.1 Experiments

For this task, the gloss for a word-sense is generated by concatenating the contexts of all training instances for that word-sense. An inverted index is generated for the glosses. The context for a test instance is fired as a query and the senses for the word are ranked using the tf-igf based cosine similarity metric described in section 3.1. The top sense is picked.

The baseline precision obtained for this task was 53.5%

The precision obtained using fine-grained scoring was 66.1% and the recall was 65.7%. The precision obtained using coarse-grained scoring was 74.3% and the recall was 73.9%.

| Gloss | ST | WS | FG | P | Precision | Recall |
|---|---|---|---|---|---|---|
| Hyper | No | 1 | T | n and v | 34.0 | 29.1 |

Table 6: Senseval-3 report

| ST | WS | FG | P | R1 | R2 |
|---|---|---|---|---|---|
| Yes | 1 | F | n | 72.9 | 88.5 |
| Yes | 1 | F | v | 43.5 | 62 |
| Yes | 1 | T | n | 65.1 | 83 |
| Yes | 1 | T | v | 26.2 | 44.07 |

Table 7: Report of Senseval-3 Extended WordNet task with modified parameters

### 6.2 Conclusion

We see that densely populated glosses do help in getting a better precision score. One possible course of action that this finding suggests is some kind of interactive WSD where the user is allowed to correct machine generated tags for some dataset. The contexts for words in the correctly tagged data could next get appended to existing gloss of the corresponding word-sense.

## References


Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine code from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM Press.

D Lin. 1998. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304, San Francisco, CA. Morgan Kaufmann.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Oliver Manson. 1980. Qtag—a portable probabilistic tagger. In *Corpus Research, The University of Birmingham, U.K.*

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography 3 (4)*, pages 235 – 244.

M. F. Porter. 1980. An algorithm for suffix stripping. In *Proceedings of SIGIR*.

Ganesh Ramakrishnan, Soumen Chakrabarthi, Deepa Paranjpe, and Pushpak Bhattacharyya. 2004. Is question answering an acquired skill ? In *Proceedings of the 13th World Wide Web Conference (WWW13)*.