JU-SKNSB: Extended WordNet Based WSD on the English All-Words Task at SemEval-1

Sudip Kumar Naskar
Computer Sc. & Engg. Dept.,
Jadavpur University,
Kolkata, India
sudip.naskar@gmail.com

Sivaji Bandyopadhyay
Computer Sc. & Engg. Dept.,
Jadavpur University,
Kolkata, India
sivaji_cse_ju@yahoo.com

Abstract

This paper presents an Extended WordNet based word sense disambiguation system using a major modification to the Lesk algorithm. The algorithm tries to disambiguate nouns, verbs and adjectives. The algorithm relies on the POS-sense tagged synset glosses provided by the Extended WordNet. The basic unit of disambiguation of our algorithm is the entire sentence under consideration. It takes a global approach where all the words in the target sentence are simultaneously disambiguated. The context includes previous and next sentence. The system assigns the default WordNet first sense to a word when the algorithm fails to predict the sense of the word. The system produces a precision and recall of .402 on the SemEval-2007 English All-Words test data.

1 Introduction

In Senseval 1, most of the systems disambiguating English words, were outperformed by a Lesk variant serving as baseline(Kilgariff & Rosenzweig, 2000). On the other hand, during Senseval 2 and Senseval 3, Lesk baselines were outperformed by most of the systems in the lexical sample track (Edmonds, 2002).

In this paper, we explore variants of the Lesk algorithm on the English All Words SemEval 2007 test data (465 instances), as well as on the first 10 Semcor 2.0 files (9642 instances). The proposed WSD algorithm is POS-sense-tagged gloss (from Extended WordNet) based and is a major modification of the original Lesk algorithm.

2 Extended WordNet

The *eXtended WordNet* (Harabagiu et al., 1999) project aims to transform the WordNet glosses into a format that allows the derivation of additional semantic and logic relations. It intends to syntactically parse the glosses, transform glosses into logical forms and tag semantically the nouns, verbs, adjectives and adverbs of the glosses automatically. The last release of the Extended WordNet is based on WordNet 2.0 and has three stages: POS tagging and parsing, logic form transformation, and semantic disambiguation.

3 Related Works

Banerjee and Pedersen (2002) reports an adaptation of Lesk's dictionary-based WSD algorithm which makes use of WordNet glosses and tests on English lexical sample from SENSEVAL-2. They define *overlap* as the longest sequence of one or more consecutive *content* words that occurs in both glosses. Each overlap contributes a score equal to the square of the number of words in the overlap.

A version of Lesk algorithm in combination with WordNet has been reported for achieving good results in (Ramakrishnan et al., 2004).

Vasilescu et al. (2004) carried on a series of experiments on the Lesk algorithm, adapted to WordNet, and on some variants. They studied the effect of varying the number of words in the contexts, centered around the target word.

But till now no work has been reported which makes use of Extended WordNet for Lesk-like gloss-oriented approach.

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 203–206,
Prague, June 2007. ©2007 Association for Computational Linguistics

4 Proposed Sense Disambiguation Algorithm

The proposed sense disambiguation algorithm is a major modification of the Lesk algorithm (Lesk, 1986). WordNet and Extended WordNet are the main resources.

4.1 Modifications to the Lesk Algorithm

We modify the Lesk algorithm (Lesk, 1986) in several ways to create our baseline algorithm. The Lesk algorithm relies on glosses found in traditional dictionaries which often do not have enough words for the algorithm to work well. We choose the lexical database *WordNet*, to take advantage of the highly inter–connected set of relations among different words that WordNet offers, and Extended WordNet to capitalize on its (POS and sense) tagged glosses.

The Lesk algorithm takes a *local approach* for sense disambiguation. The disambiguation of the various words in a sentence is a series of independent problems and has no effect on each other. We propose a *global* approach where all the words (we mean by *word*, an open-class lemma) in the context window are *simultaneously* disambiguated in a bid to get the best *combination* of senses for all the words in the window instead of only the target word. The process can be thought of as sense disambiguation of the whole context, instead of a word.

The Lesk algorithm disambiguates words in short phrases. But, the basic unit of disambiguation of our algorithm is the entire sentence under consideration. We later modify the context to include the previous and next sentence.

Another major change is that the dictionary definition or gloss of each of its senses is compared to the glosses of every other word in the context by the Lesk algorithm. But in the present work, the words themselves are compared with the glosses of every other word in the context.

4.2 Choice of Which Glosses to Use

While Lesk's algorithm restricts its comparisons to the dictionary meanings of the words being disambiguated, our choice of dictionary allows us to also compare the meanings (i.e., glosses) of the words, as well as the words that are related to them through various relationships defined in WordNet. For each POS we choose a relation if links of its kind form at least 5% of the total number of links for that part of speech, with two exceptions. We use the *attribute* relation although there are not many links of its kind. But this relation links adjectives, which are not well developed in WordNet, to nouns which have a lot of data about them. This potential to tap into the rich noun data prompted us to use this relation. Another exception is the *antonymy* relationship. Although there are sufficient antonymy links for adjectives and adverbs, we have not utilized these relations.

Noun	Verb	Adjective
Hypernym	Hyponym	Attribute
Hyponym	Troponym	Also see
Holonym	Also see	Similar to
Meronym		Pertainym of
Attribute		

Table 1. WordNet relations chosen for the disambiguation algorithm

4.3 The Algorithm

The gloss bag is constructed for every sense of every word in the sentence. The gloss-bag is constructed from the POS and sense tagged glosses of synsets, obtained from the Extended WordNet. For any synset, the words forming the synset and the gloss definition contribute to the gloss-bag. The non-content words are left out. Example sentences do not contribute to the gloss bag since they are not (POS and sense) tagged. Each word along with its POS and sense-tag are stored in the gloss bag. For words with different POS, different relations are taken into account (according to Table 1) for building the corresponding gloss-bag.

This gloss-bag creation process can be performed offline or online. It can be performed dynamically on a as-when-needed basis. Or, gloss-bags can be created for all WordNet entries only once and stored in a data file in prior. The issue is time versus space.

Once, this gloss-bag creation process is over, the comparison process starts. Each word (say W_i) in the context is compared with each word in the gloss-bag for every sense (say S_k) of every other word (say W_j) in the context. If a match is found, they are checked further for part-of-speech match. If the words match in part-of-speech as well, a score is assigned to both the words: the word being matched (W_i) and the word whose gloss-bag contains the match (W_j). This matching event indicates

mutual confidence towards each other, so both words are rewarded for this event. Two two-dimensional (one for word index and the other for sense index) vectors are maintained: *sense_vote* for the word in context, and *sense_score* for the word in gloss-bag. Say, for example, the context word (W_i # *noun*) matches with gloss word (W_n # *noun* # *m*) (i.e., $W_i = W_n$) in the gloss bag for k^{th} sense of W_j. Then, a score of 1/(gloss bag size of (W_{jk})) is assigned to both *sense_vote*[i][m] and *sense_score*[j][k]. Scores are normalized before assigning because of huge discrepancy in gloss-bag sizes. This process continues until each context word is matched against all gloss-bag words for each sense of every other context words.

Once all the comparisons have been made, we add *sense_vote* value with the *sense_score* linearly value for each sense of every word to arrive at the combination score for this word-sense pair.

The algorithm assigns a word the n^{th} sense for which the corresponding *sense_vote* and *sense_score* produces the maximum sum, and it does not assign a word any sense when the corresponding *sense_vote* and *sense_score* values are 0, even if the word has only one sense. In the event of a tie, we choose the one that is more frequent, as specified by WordNet.

Assuming that there are N words in the window of context (i.e. the sentence), and that, on an average there are S senses per word, and G number of gloss words in each gloss bag per sense, $N * S$ gloss bags need to be constructed, giving rise to a total of $N * S * G$ gloss words. Now these many gloss words are compared against each of the N context words. Thus, $N^2 * S * G$ pairs of word comparisons need to be performed. Both, S and G vary heavily.

5 Variants of the Algorithm

The algorithm discussed thus far is our baseline algorithm. We made some changes, as described in the following two subsections, to investigate whether the performance of the algorithm can be improved.

5.1 Increasing the Context Size

The poor performance of the algorithm perhaps suggests that sentential context is not enough for this algorithm to work. So we went for a larger context: a context window containing the current sentence under consideration (target sentence), its preceding sentence and the succeeding sentence. This increment in context size indeed performed better than the baseline algorithm.

5.2 Assigning Different Scores

When constructing the gloss-bags for a word-sense pair, some words may appear in more than one gloss (by gloss we mean to say synonyms as well as gloss). So, we added another parameter with every (word#pos#sense) in a gloss bag: *noc* - the number of occurrence of this (word#pos#sense) combination in this gloss-bag.

And, in case of a match of context word (say W_i) with a gloss-bag word (of say k^{th} sense of word W_j), we scored the words in four ways to see if this phenomenon has any effect on the sense disambiguation process. Say, for example, the context word (W_i # *noun*) matches with gloss word (W_n # *noun* # *m* # *noc*) in the gloss bag for k^{th} sense of W_j (i.e., the particular word appears *noc* times in the said gloss-bag) and the gloss bag size is *gbs*. Then, we reward W_i and W_j for this event in four ways given below.

1. Assign 1/*gbs* to *sense_vote*[i][m] and 1/*gbs* to *sense_score*[j][k].
2. Assign 1/*gbs* to *sense_vote*[i][m] and *noc*/*gbs* to *sense_score*[j][k].
3. Assign *noc*/*gbs* to *sense_vote*[i][m] and 1/*gbs* to *sense_score*[j][k].
4. Assign *noc*/*gbs* to *sense_vote*[i][m] and *noc*/*gbs* to *sense_score*[j][k].

The results of this four-way scoring proved that this indeed has influence on the disambiguation process.

The WSD system is based on Extended WordNet version 2.0-1.1 (the latest release), which is in turn based on WordNet version 2.0. So, the system returns WordNet 2.0 sense indexes. These WordNet sense indexes are then mapped to WordNet 2.1 sense indexes using sensemap 2.0 to 2.1.

6 Evaluations

The system has been evaluated on the SemEval-2007 English All-Words Tasks (465 test in-

stances), as well as on the first 10 Semcor 2.0 files, which are manually disambiguated text corpora using WordNet senses.

We compute *F-Score as 2*P*R / (P+R)*. Table 2 shows the performance of the four variants of the system (with a context size of 3 sentences) on the first 10 Semcor 2.0 files. From table 2, it is clearly evident that model C produces the best result (precision - .621, recall - .533) among the 4 scoring schemes. POS-wise evaluation results for model C on Semcor 2.0 data is given in table 3.

	Model			
	A	B	C	D
Precision	.618	.602	.621	.604
Recall	.531	.517	.533	.519
F-Score	.571	.556	.574	.558

Table 2. Evaluation of the four models on Semcor Data

	Noun	Verb	Adj	Overall
Precision	.6977	.4272	.6694	.6211
Recall	.6179	.3947	.4602	.5335
F-Score	.6554	.4103	.5454	.574

Table 3. POS-wise Evaluation for model C on Semcor Data

Model C produced a precision of .393 and a recall of .359 on the SemEval-2007 English All-Words test data (465 test instances). Table 4 shows POS-wise evaluation results for this test data.

	Noun	Verb	Overall
Precision	.507	.331	.393
Recall	.472	.299	.359
F-Score	.489	.314	.375

Table 3. POS-wise Evaluation on SemEval-2007 English All-Words test data

When default WordNet first senses were assigned to the (40) words for which the algorithm failed to predict senses, both the precision and recall values went up to .402 (this result has been submitted in SemEval-2007). The WSD system stood 10th in the SemEval-2007 English All-Words task.

7 Discussions

We believe that this somewhat poor showing can be partially attributed to the brevity of definitions in WordNet in particular and dictionaries in general. The Lesk algorithm is crucially dependent on the lengths of glosses. However lexicographers aim to create short and precise definitions which, though a desirable quality in dictionaries, is disadvantageous to this algorithm. Nouns have the longest average glosses in WordNet, and indeed the highest recall obtained is on nouns. The characteristics of the gloss bags need to be further investigated. Again many of the sense tagged gloss words in Extended WordNet, which are determinant factors in this algorithm, are of "silver" or "normal" quality. And finally, since the system returns WordNet 2.0 sense indexes which are mapped to WordNet 2.1 indexes with certain amount of confidence using sensemap 2.0 to 2.1, there may be some loss of information during this mapping process.

References

A. Kilgarriff, and J. Rosenzweig. 2000. Framework and Results for English SENSEVAL. *Computers and the Humanities*, 34, 15-48.

Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. 2004. Evaluating Variants of the Lesk Approach for Disambiguating Words. *LREC*, Portugal.

G. Ramakrishnan, B. Prithviraj, and P. Bhattacharyya. 2004. A Gloss Centered Algorithm for Word Sense Disambiguation. *Proceedings of the ACL SENSEVAL 2004*, Barcelona, Spain, 217-221.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. *Proceedings of SIGDOC '86*.

P. Edmonds. 2002. SENSEVAL : The Evaluation of Word Sense Disambiguation Systems, *ELRA Newsletter*, Vol. 7, No. 3.

S. Banerjee. 2002. Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet. *MS Thesis*, University of Minnesota.

S. Banerjee, and T. Pedersen. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *CICLing*, Mexico.

S. Harabagiu, G. Miller, and D. Moldovan. 1999. WordNet2 - a morphologically and semantically enhanced resource. *Proceedings of SIGLEX-99*, Univ of Mariland. 1-8.