*Computers and the Humanities* **34:** 147–152, 2000.



# Word Sense Disambiguation with a Similarity-Smoothed Case Library

DEKANG LIN
*Department of Computer Science, University of Manitoba, Canada*
*(E-mail: lindek@cs.umanitoba.ca)*

## 1. Introduction

We present a case-based algorithm for word sense disambiguation (WSD). The case library consists of local contexts of sense-tagged examples in the training corpus. For each target word in the testing corpus, we compare its local context with all known cases and assign it the same sense tag as in the most similar case. Like other corpus-based WSD algorithms, data sparseness is a serious problem. In order to alleviate this problem, an automatically generated thesaurus is employed that allows a match between two local contexts to be established even when different words are involved.

## 2. Representation of Local Context

In many WSD algorithms, the local context of a word is mainly made up of the words surrounding the word. In our approach, the local context of a word is a set of paths in the dependency tree of the sentence that contains the word.

The nodes in the dependency tree of a sentence represent words in the sentence. The links represent the dependency relationships between the words. A dependency relationship is an asymmetric binary relationship between two words: the **head** (or **governor**) and the **modifier** (or **dependent**). The properties of the smallest phrase that contains both the head and the modifier are mostly determined by the head. For example, the dependency tree of the sentence (1a) is shown in (1b).

(1) a. Ethnic conflicts are shaking the country

b.

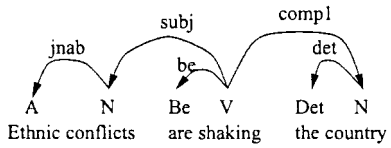


*Table I.* Meanings of dependency labels.

| Label | Meaning |
|---|---|
| Compl | the relationship between a word and its first complement |
| det | the relationship between a noun and its determiner |
| jnab | the relationship between a noun and its adjectival modifier |
| subj | the relationship between a subject and a predicate |
| gen | the relationship between a noun and its genitive determiner |
| rel | the relationship between a noun and its relative clause |

The root node of the dependency trees is "shaking". The arrows of the links point to the modifiers. The labels attached to the links are the types of dependency relationships. Explanations of the labels can be found in Table I.

We define the local context of a word in a sentence to be a set of paths in the dependency tree of the sentence between the word and other words in the sentence. Each path is a feature of the word. The features are named by concatenating the link labels and part-of-speech tags of the nodes along the paths. The value of a feature is the root form of the word at the other end of the path. The set of feature-value pairs forms the local context of the word. For example, the local context of "shaking" in (la) is the feature vector (2).

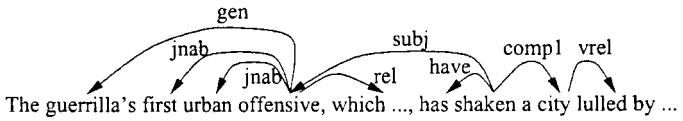
(2) ((V shake), (V:subj:N conflict), (V:subj:N:jnab:A ethnic),
(V:be:Be be), (V:compl:N country), (V:compl:N:det:Det the))

We used a broad-coverage parser, called Principar (Lin, 1993), to parse all the training examples and extract the local context of the sense-tagged words. The local contexts of target words in the testing corpus are similarly constructed. The intended meaning of a target word is determined by finding the sense-tagged example whose local context is most similar to the local context of the target word.

## 3. Similarity Measure

To deal with the data sparseness problem, we used a thesaurus automatically extracted from a large corpus (125 million words) to bridge the gap between the training examples and the testing corpus (Lin, 1998). Consider the example (3a) from the Senseval testing corpus. The relevant part of its dependency tree is shown in (3b). The local context of "shaken" is the feature vector in (4).

(3) a. The guerrillas' first urban offensive, which has lasted three weeks so far and shows no sign of ending, has shaken a city lulled by the official propaganda.

b. 


(4) ((V shake), (V:subj:N offensive), (V:compl:N city), . . . )

Compared with the example in (1), the subject and objects of "shake" in (3) are different words. However, by looking up the automatically generated thesaurus, which contains 11,870 noun entries, 3,644 verb entries and 5,660 adjective/adverb entries, our system found the following entries for "offensive" and "conflict":

**offensive:** attack 0.183; assault 0.168; raid 0.154; effort 0.153; campaign 0.148; crackdown 0.137; strike 0.129; bombing 0.127; move 0.124; invasion 0.123; initiative 0.121; . . . conflict 0.072; . . .

**city:** state 0.346; town 0.344; country 0.299; country 0.292; university 0.286; region 0.248; village 0.237; area 0.228; . . .

The similarity between "city" and "country" is 0.292 and the similarity between "offensive" and "conflict" is 0.072. The similarities between these words enable the system to recognize the commonality between the local context (4) and (2). If all distinct words were considered as equally different, the sentence "she shook her head" would have as much commonality to (3a) as (1a), which is that the main verb is "shake".

Let $v$ be a feature vector and $f$ be a feature. We use $l(f)$ to denote the length of the path that corresponds to $f$, $F(v)$ to denote the set of features in $v$ and $f(v)$ to denote the value of feature $f$ in $v$. For example, suppose $v$ is the feature vector in (2) and $f$ is the feature V:subj:N, then $l(f) = 1$, $f(v)$ is "conflict" and $F(v) = \{$V, V:subj:N, V:subj:N:Jnab:A, V:be:Be, V:compl:N, V:compl:N:det:Det$\}$.

The function simTo($v_1$, $v_2$) measures the similarity of $v_1$ to $v_2$. It is defined as follows:

$$\frac{\sum f \in F(v_1) \cap F(v_2) 3^{-1(f)} sim(f(v_1)), f(v_2))(r log P(f(v_1)) + log P(f(v_2)))}{r \sum f \in F(v_1) 3^{-l(f)} log P(f(v_1)) + \sum f \in F(v_2) 3^{-l(f)} log P(f(v_2))}$$

where $r \in [0, 1]$ is a discount factor to make simTo($V_1$, $v_2$) asymmetrical; sim($w$, $w'$) is the similarity between two words $w$ and $w'$, retrieved from the automatically generated thesaurus; $P(f(v))$ is the prior probability of the value of feature $f$ of the verb $v$. Suppose, $v$ is the verb "shaking" in (1b), $f$ is the feature V:subj:N. Then $f(v)$ is [$_N$ conflict] and $P(f(v))$ is estimated by dividing the frequency of [$_N$ conflict] in a large corpus with the total number of words in the corpus.

The value $3^{-l(f)}$ is used in simTo($v_1$, $v_2$) to capture the fact that the longer the path, the smaller the influence that the word at the other end can exert on the target word.

Examples in the training corpus often contain irrelevant details that have nothing to do with the meaning of the target word. The feature (V:be:Be be) in (2) is one such example. The decision process should focus more on how much of the unknown instance is covered by a known case. This is achieved by using the discount factor $r$ (set to 0.1 in all our experiments) to make simTo ($v_1$, $v_2$) asymmetrical. The value simTo($v_1$, $v_2$) is high when $v_1$ possesses the most of the features of $v_2$. Extra features in $v_1$ that are not shared by $v_2$ are discounted by $r$.

Given a target word and its local context $v$, our algorithm tags the target word with the sense tag of the example whose local context $v\prime$ maximizes the similarity simTo($v\prime$, $v$).

## 4. Experimental Results

We submitted two sets of results to the Senseval workshop. The first one used the entire training corpus to construct the case library. In the second one, the case library contains only the examples from the Hector lexicon. Our official Senseval results are as follows:

| | | |
|---|---|---|
| Trained with the corpus: | recall=.701, | precision=.706 |
| Trained with the lexicon: | recall=.520, | precision=.523 |

All evaluation results reported in this paper are obtained with the "Coarse Grain" scoring algorithm.

Our official system had several serious bugs, which were later corrected. Table II shows our unofficial results after the bug fixes. The column caption "R" stands for recall, "P" stands for precision and "F" stands for the F-measure, which is defined as $F = \frac{2 \times P \times R}{P+R}$. Table II includes the results of several variations of the system that we experimented with:

To gauge the effect of the amount of training data on WSD, we constructed a case library with the training corpus and another one with the examples from the Hector lexicon.

To see the advantage of the thesaurus, we also ran the system without it. The thesaurus accounted for about 4–6% increase in both precision and recall. It is somewhat surprising that the benefits of the thesaurus is not greater with the smaller training set than with the larger one.

To determine how the similarity of cases affects the reliability of the disambiguation decisions, we used a threshold $\theta$ to filter the system outputs. The system only assigns a sense tag to a word when the similarity of the most similar case is greater than $\theta$. Table II shows that a low threshold seems to produce slight improvements. A high threshold causes the recall to drop drastically with only modest gain in precision.

*Table II.* Unofficial evaluation results.

| Using paths in dependency tree as features | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| use | training | $\theta = 0$ | | | $\theta = 0.25$ | | | $\theta = 0.5$ | | |
| Thesaurus | data | R | P | F | R | P | F | R | P | F |
| no | corpus | .698 | .692 | .695 | .687 | .702 | .694 | .622 | .728 | .670 |
| yes | corpus | .748 | .754 | .750 | .733 | .771 | .751 | .684 | .781 | .729 |
| no | lexicon | .587 | .596 | .591 | .578 | .598 | .588 | .438 | .633 | .518 |
| yes | lexicon | .628 | .637 | .632 | .614 | .650 | .631 | .541 | .663 | .596 |
| Using surrounding words as features | | | | | | | | | | |
| use | training | $\theta = 0$ | | | $\theta = 0.25$ | | | $\theta = 0.5$ | | |
| Thesaurus | data | R | P | F | R | P | F | R | P | F |
| no | corpus | .623 | .628 | .625 | .589 | .641 | .613 | .279 | .762 | .408 |
| yes | corpus | .671 | .678 | .674 | .377 | .787 | .510 | .121 | .873 | .213 |
| no | lexicon | .462 | .466 | .464 | .370 | .458 | .409 | .082 | .711 | .147 |
| yes | lexicon | .506 | .512 | .509 | .143 | .741 | .240 | .029 | .810 | .056 |

To evaluate the contribution of parsing in WSD, we experimented with a version of the system which uses surrounding words and their part-of-speech tags as features. For example, the feature vector for sentence (1a) is:

((V shake) (prev3:A ethnic) (prev2:N conflict) (prev1:Be be)
next1:Det the) (next2:Det city))

The use of the parser leads to about 7% increase in both recall and precision when the training corpus is used and about 12% in both recall and precision when only the Hector examples are used.

## 5. Related Work

Many recent WSD algorithms are corpus-based (e.g., Bruce and Wiebe, 1994; Ng and Lee, 1996; and Yarowsky, 1994), as well as most systems described in this special issue. Leacock and Chodorow (1998) explored the idea of using WordNet to deal with the data sparseness problem. They observed that as the average number of training examples per word sense is increased from 10 to 200, the improvement in the accuracy (roughly equivalent to the precision measure in Senseval) gained by the use of WordNet decreases from 3.5% to less than 1%. In our experiments, however, the improvement in precision gained by the use of the automatically generated thesaurus increases from 5.2% to 6.9% ($\theta = 0.25$) as the average number of examples per sense is increased from 3.67 (in Hector) to 30.32 (in the training corpus).

## 6. Conclusion

We presented a case-based algorithm for word sense disambiguation. Our results with the Senseval data showed that the use of the automatically generated thesaurus significantly improves accuracy of WSD. We also showed that defining local contexts in terms of dependency relationships has substantial advantage over defining local contexts as surrounding words, especially when the size of the training set is very small.

## References

Bruce, R. and J. Wiebe. 'Word sense disambiguation using decomposable models'. *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*. Las Cruces, New Mexico, 1994, pp. 139–145.

Leacock, C. and M. Chodorow. 'Combining Local Context and WordNet Similarity for Word Sense Identification'. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. pp. 256–283.

Lin, D. 'Principle-based Parsing without Overgeneration'. *Proceedings of ACL–93*. Columbus, Ohio, 1993, pp. 112–120.

Lin, D.: 'Automatic Retrieval and Clustering of Similar Words'. *Proceedings of COLING/ACL–98*. Montreal, 1998, pp. 768–774.

Ng, H. T. and H. B. Lee. 'Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Examplar-Based Approach'. *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics*. Santa Cruz, California, 1996, pp. 40–47.

Yarowsky, D. 'Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French'. *Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico, 1994, pp. 88–95.