Computers and the Humanities **34:** 165–170, 2000.

Simple Word Sense Discrimination

Towards Reduced Complexity

KEITH SUDERMAN
Department of Computer Science, University of Manitoba, Winnipeg, Canada R3T 2N2
(E-mail: suderman@cs.umanitoba.ca)

Abstract. Wisdom is a system for performing word sense disambiguation (WSD) using a limited number of linguistic features and a simple supervised learning algorithm. The most likely sense tag for a word is determined by calculating co-occurrence statistics for words appearing within a small window. This paper gives a brief description of the components in the Wisdom system and the algorithm used to predict the correct sense tag. Some results for Wisdom from the Senseval competition are presented, and directions for future work are also explored.

Key words: Senseval, statistical WSD, word sense disambiguation

1. Introduction

For any non-trivial problem in computer science, reducing complexity is an important goal. As the problems become more difficult the complexity of solutions tends to increase. Word Sense Disambiguation (WSD) is a non-trivial task, and as the sophistication of the systems that perform WSD increases, the complexity of these systems also increases. Unfortunately, this increase in complexity is frequently exponential rather than linear or (ideally) logarithmic.

This paper describes Wisdom, a WSD system developed for a graduate level course in Natural Language Understanding (NLU) and then expanded to take part in the Senseval competition.[1] The initial Wisdom system was an attempt to study the predictive power of co-occurrence statistics without considering other linguistic features. To select a sense tag, the initial system calculated co-occurrence statistics for words within a four-word window. Larger windows were tested; however, the best results were achieved across all words when a small word window is used. This agrees with past observations by Kaplan (1955), Choueka and Lusignan (1995), and others, that humans require only a two-word window to distinguish the correct sense of a word. For the Senseval exercise, Wisdom was augmented to construct a dependency tree for the context sentence and consult a thesaurus to overcome sparse training data.

Wisdom performs very well considering the limited amount of knowledge employed, achieving an overall fine-grained precision of 69.0% with 60.8% recall

on 7,444 words attempted. Only the English language tasks were tested, but the system can be trained with a tagged corpus in any language.

2. Statistical Word Sense Disambiguation

Wisdom can disambiguate any word ω for which a previously tagged corpus $\mathcal{S}$ is available. The task of assigning sense tags to the occurrences of ω in an untagged corpus $\mathcal{T}$ is divided into two phases, a training phase and a classification phase. During the training phase *relevant* words are extracted from the sentence $\mathcal{S}$ and a count of the number of times they occur with each possible sense of the word ω is maintained. After the sentences in $\mathcal{S}$ have been examined and relevant words counted, the sentences in $\mathcal{T}$ are presented and each occurrence of ω is sense-tagged. Identification of relevant words is discussed in detail in the next section.

2.1. RELEVANT WORDS AND PHRASES

Initially, relevant words are considered to be those words immediately adjacent to ω in the context sentence. Empirical testing suggests that only the two words immediately preceding ω and the two words immediately following ω should be considered, including function words and other common stop words. For example, for the adjective *generous* in the sentence:

"They eat reasonably *generous* meals and they snack in between."

eat, *reasonably*, *meals*, and *and* are considered to be relevant words. In addition to maintaining occurrence counts for single relevant words, frequencies for combinations of adjacent words are also computed to enable recognition of commonly occurring phrases. If the word ω appears in the phrase "$u\ v\ \omega\ x\ y$" then frequency statistics are also maintained for the strings uv, vx, xy, and $uvxy$. These are referred to as *relevant phrases*.

2.2. TRAINING

During the training phase the sentences in $\mathcal{S}$ are parsed, the position of the word ω is determined, relevant words and phrases are identified, and the number of times each relevant word or phrase co-occurs with ω is counted. After all relevant words have been recorded, the occurrence counts are converted to conditional probabilities $P(i|r)$, that is:

$$p_i = \frac{r_i}{\sum_{j=1}^{n} r_j}$$

where r_i is the number of times the relevant word r has appeared with sense i, and n is the number of possible sense tag assignments to ω. This yields the probability that ω is an occurrence of sense i given the relevant word r.

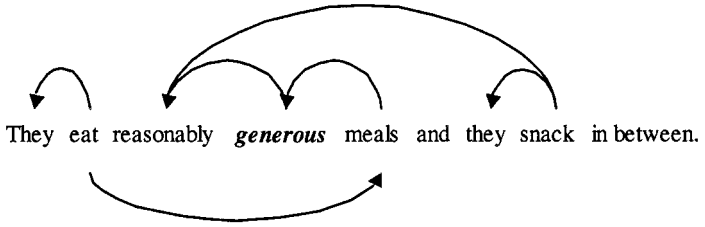

Figure 1. Dependency tree produced by Minipar.

After parsing the sentences in the training set, the Hector dictionary is searched for special cases of the word ω. A special case is a word, compound word, or morphological form of a word that has only one possible sense assignment. For example, waist *band*, steel *band*, and t-*shirt* all appear in the dictionary with unique sense tags, while *wooden* spoon, and wrist*band* have two possible sense tags and are not, therefore, considered as special cases. Sense tags are assigned to special cases by performing a dictionary lookup and assigning the indicated sense.

It should be noted that morphological forms of the word ω are treated separately as distinct words rather than as different forms of the same word. This is an artifact of the original system that used a simple tokenizer, rather than fully parsing the sentence.

After the training phase and before classification, entropy values are calculated for co-occurring words, and all those with entropy above a predetermined threshold are considered poor sense indicators for ω and subsequently ignored. Entropy is calculated for word r as:

$$\text{entropy} = \sum_{i=1}^{n} -v_i \times \text{Log}_2(v_i)$$

where v_i is the conditional probability $P(sense_i|r)$, and n is the number of possible sense assignments to ω. The threshold used to determine whether a relevant word is ambiguous depends on ω, as well as other factors such as the size and source of the corpus. The system that participated in Senseval simply used the same entropy threshold for all words.

2.3. ADDITIONAL SOURCES OF KNOWLEDGE

If the size of the training set is small, the number of reliable indicators may be insufficient to identify infrequently occurring senses. In such cases, Wisdom uses two additional knowledge sources: First, sentences are parsed with Minipar (Lin, 1993; 1998), a broad coverage parser for English. Minipar generates a dependency tree for each word in the sentence that specifies the head of the phrase in which it occurs. For example, for the above sentence Minipar generates the dependency tree shown in Figure 1.

The dependency tree is used to identify the phrase containing the word ω. Relevant words are restricted to adjacent words in the same phrase in the target sentence. For the above example, the relevant words are *reasonably*, *meals*, *eat*, and *they*. Since parsing with Minipar is a recent addition to the system, this is the only information provided by Minipar that is currently used by Wisdom, although there are clearly possibilities for enhancing the system with additional information from the parse.

While the use of dependency trees improves the quality of the relevant words, it does not overcome the problem of a small training set. Therefore, during classification, if none of the relevant words has been previously encountered Wisdom consults an electronic thesaurus (Lin, 1998) to find words similar to the relevant words. Each of these is assigned a similarity value by the thesaurus and words above a predetermined threshold are retained.

2.4. CLASSIFICATION

After training, sentences from the test set are presented to the system one at a time for classification, and the relevant words are extracted. The conditional probabilities for relevant words that have been encountered in the training set are summed, and ω is tagged with the sense that has the highest sum of probabilities. If there is more than one possible sense assignment, one is chosen at random.

If the system is unable to determine a possible sense assignment, it will attempt to guess the correct sense tag. The sense to be used as a guess is determined during training. A set of 100 trial runs is performed for each possible sense tag. In each set of runs a different sense is used as the default guess: the first sense is used in the first set, the second sense is used in the second set, etc. During each trial run a portion of the training set is drawn at random and presented to the system for training. The remainder of the training set is classified and the score is recorded. The sense that yields the best average score is used as the default guess when classifying the hold-out data. Interestingly, the most frequently occurring sense is rarely the best sense to select when there are no other cues, since if the training set is sufficiently large there is typically some evidence (in the form of previously encountered relevant words) for the most frequently occurring senses. Therefore, when no relevant words are found, we may assume that this is an instance of a less frequently occurring sense of ω. Use of this information in Wisdom is currently under exploration.

3. Results

The results presented here are those from the September competition. No results were submitted for the second evaluation in October. There are still several obvious problems with the system, which are currently under investigation. For example, Wisdom attempted to assign sense tags to five more verbs than the human

Table I. Overall score for All-trainable words

	Precision	Recall	Attempted	Position
Fine grain	69.0	60.8	7044	5
Mixed grain	71.8	63.3	7444	6
Coarse grain	73.8	65.0	7444	7

Table II. Fine (Coarse)-grained scores by part of speech

	Precision	Recall	Attempted	Position
Nouns	73.4 (79.6)	56.4 (61.2)	2914	6 (7)
Verbs	64.3 (68.3)	64.2 (68.2)	2904	6 (7)
Adjectives	72.1 (76.4)	65.9 (69.8)	1284	5 (4)

annotators, which indicates either an incorrect part of speech tagging by the parser or a problem in Wisdom itself.

Table I shows the overall system performance for all trainable words, Table II shows system performance by part of speech. In relation to other systems, Wisdom performed better than expected, typically finishing in the top five to ten systems for all tasks, and performing slightly better on adjectives than nouns or verbs. While Wisdom's coarse-grained scores tended to be higher than its fine-grained scores, Wisdom's coarse-grained scores did not increase as much as other systems and typically fell behind when compared to the other systems on course-grained sense distinction. However, for all trainable adjectives, Wisdom achieved the fifth highest fine-grained score and the fourth highest coarse-grained score.

4. Future Work

Wisdom represents a first attempt to develop a system for WSD. The original system was developed for a graduate level AI course and was not intended to be extended; however, performance of the system in the Senseval exercise, especially given the simplicity of the system's design, suggests it may be worthwhile to continue to improve the system.

In particular, because Wisdom is a relatively simple system, it should be possible to develop Wisdom in such a way as to enable a systematic study of the contribution of different types of information to the disambiguation task. Currently, most systems employ various kinds of contextual and external information (see Ide and Véronis (1998) for a comprehensive survey). Typically, the contribution of each type of information, especially for disambiguating words in different parts of speech etc., is difficult or impossible to determine, and no systematic study has, to

my knowledge, yet been conducted. However, given the complexity of WSD, such a study could shed light on some of the subtleties involved.

To accomplish this, baseline performance levels need to be firmly established for the system in its current state before other sources of knowledge are added. The results from the Senseval competition need to be studied in detail to determine what, if any, relation exists between the words Wisdom can correctly tag and those it cannot. In addition, parameters need to be tailored specifically to the target word rather than using one set of global parameters across all words. Finally, the relation between the choice of parameters and word classes will also be investigated. Once solid baselines have been established for the system, other sources of linguistic knowledge can be added. In particular, the parser provides much more information than is used.

Note

[1] Wisdom appears as `manitoba.ks` in the Senseval results.

References

Choueka, Y. and S. Lusignan. "Disambiguation by Short Contexts". *Computers and the Humanities*, 19 (1985), 147–157.

Ide, N. and J. Véronis. "Word Sense Ambiguation: The State of the Art". *Computational Linguistics*, 24(1) (1998), 1–40.

Kaplan, A. "An Experimental Study of Ambiguity and Context". *Mechanical Translation*, 2(2) (1955), 39–46.

Lin, D. "Principle Based Parsing without Overgeneration". In *31st Annual Meeting of the Association for Computational Linguistics*, Columbus Ohio, 1993, pp. 112–120.

Lin, D. "Automatic Retrieval and Clustering of Similar Words". In *COLING-ACL98*, Montreal, Canada, 1998.