Simple Features for Statistical Word Sense Disambiguation

Abolfazl K. Lamjiri, Osama El Demerdash, Leila Kosseim

CLaC Laboratory
Department of Computer Science
Concordia University, Montreal, Canada
{a_keigho,osama_el,kosseim}@cs.concordia.ca

Abstract

In this paper, we describe our experiments on statistical word sense disambiguation (WSD) using two systems based on different approaches: Naïve Bayes on word tokens and Maximum Entropy on local syntactic and semantic features. In the first approach, we consider a context window and a sub-window within it around the word to disambiguate. Within the outside window, only content words are considered, but within the sub-window, all words are taken into account. Both window sizes are tuned by the system for each word to disambiguate and accuracies of 75% and 67% were respectively obtained for coarse and fine grained evaluations. In the second system, sense resolution is done using an approximate syntactic structure as well as semantics of neighboring nouns as features to a Maximum Entropy learner. Accuracies of 70% and 63% were obtained for coarse and fine grained evaluations.

1 Introduction

In this paper, we present the two systems we built for our first participation in the English lexical sample task at Senseval-3. In the first system, a Naïve Bayes learner based on context words as features is implemented. In the second system, an approximate syntactic structure, in addition to semantics of the nouns around the ambiguous word are selected as features to learn with a Maximum Entropy learner.

In Section 2, a brief overview of related work on WSD is presented. Sections 3 and 4 provide specifications of our two systems. Section 5 discusses the results obtained and remarks on them, and finally in Section 6, conclusion and our future work direction are given.

2 Related Work

In 1950, Kaplan carried out one of the earliest WSD experiments and showed that the accuracy of sense resolution does not improve when more than four words around the target are considered (Ide and Véronis, 1998). While researchers such as Masterman (1961), Gougenheim and Michea (1961), agree with this observation (Ide and Véronis, 1998), our results demonstrate that this does not generally apply to all words. A large context window provides *domain information* which increases the accuracy for some target words such as *bank.n*, but not others like *different.a* or *use.v* (see Section 3). This confirms Mihalcea's observations (Mihalcea, 2002). In our system we allow a larger context window size and for most of the words such context window is selected by the system.

Another trend consists in defining and using semantic preferences for the target word. For example, the verb *drink* prefers an animate subject in its *imbibe* sense. Boguraev shows that this does not work for polysemous verbs because of metaphoric expressions (Ide and Véronis, 1998).

Furthermore, the grammatical structures the target word takes part in can be used as a distinguishing tool: "the word 'keep', can be disambiguated by determining whether its object is gerund (He kept eating), adjectival phrase (He kept calm), or noun phrase (He kept a record)" (Reifler, 1955). In our second system we approximate the syntactic structures of a word, in its different senses.

Mooney (Mooney, 1996) has discussed the effect of bias on inductive learning methods. In this work we also show sensitivity of Naïve Bayes to the distribution of samples.

3 Naïve Bayes for Learning Context Words

In our approach, a large window and a smaller sub-window are centered around the target word. We account for all words within the sub-window but use a POS filter as well as a short stop-word list to filter out non-content words
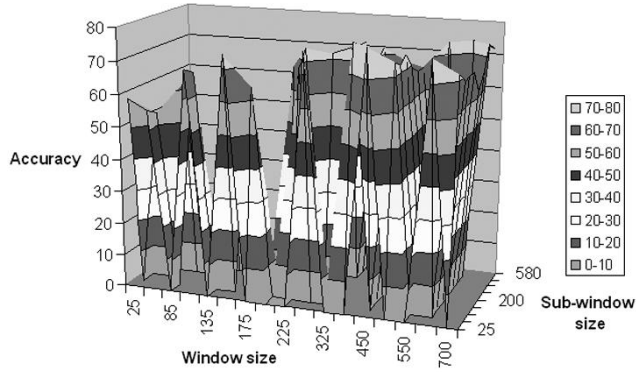
Figure 1: The effect of choosing different window and sub-window sizes for the word *bank.n.* The best accuracy is achieved with a window and sub-window size of around 450 and 50 characters respectively, while for example 50 and 25 provide very low accuracy.

from the context. The filter retains only open class words, i.e. nouns, adjectives, adverbs, and verbs, and rejects words tagged otherwise.

3.1 Changing the context window size

Figure 1 shows the effect of selecting different window and sub-window sizes for the word *bank.n.* It is clear that precision is very sensitive to the selected window size. Other words also have such variations in their precision results.

The system decides on the best window sizes for every word by examining possible window size values ranging from 25 to 750 characters[1]. Table 1 shows the optimal window sizes selected for a number of words from different word classes. The baseline is considered individually for every word as the ratio of the most common sense in the training samples. We used the Senseval-3 training set for the English lexical sample task for training. It includes a total of 7860 tagged samples for 57 ambiguous words. 15% of this data was used for validation, while the rest was used for training.

3.2 Approximate Smoothing

During the testing phase, given the context of the target word, the score of every sense is computed using the Naïve Bayes formula:

$$Score_{sense_i} = log\ p(sense_i) + \sum_k log\ p(word_k)$$

where, $word_k$ is every word inside the context window (recall that these are all the words in the sub-window, and filtered words in the large window).

Various smoothing algorithms could be used to reduce the probability of seen words and distributing them among unseen words. However, tuning various smoothing parameters is delicate as it involves keeping an appropriate amount of held-out data. Instead, we implemented an approximate smoothing method, which seems to perform better compared to Ng's (Ng, 1997) approximate smoothing. In our simple approximate smoothing the probability of seen words is not discounted to compensate for those of unseen words[2]. Finding a proper value to assign to unseen words was done experimentally; for a relatively large training data set, $p(an\ unseen\ word) = 10^{-10}$ and for a small set, 10^{-9} resulted in the highest accuracy with our 15% validation set[3]. The intuition is that, with a small training set, more *unseen* words are likely to be seen during the testing phase, and in order to prevent the accumulating score penalty value from becoming relatively high, a lower probability value is selected. Additionally, the selection should not result in large differences in the computed scores of different senses of the target word.

A simple function assigns 10^{-10} in any of the following conditions: the total number of words seen is larger than 4300, the number of training instances is greater than 230, or the context window size is larger than 400 characters. The function returns 10^{-9} otherwise.

4 Maximum Entropy learning of syntax and semantics

Syntactic structures as well as semantics of the words around the ambiguous word are strong clues for sense resolution in many words. However, deriving and using exact syntactic information introduces its own difficulties. So, we tried to use approximate syntactic structures by learning the following features in a context window *bounded by the last punctuation before and the first punctuation after* the ambiguous word:

1. *Article Bef:* If there is any article before, the string token is considered as the value of this feature.

2. *POS, POS Bef, POS Aft:* The part of speech of the target, the part of speech of the word before (after) if any.

---

[1] For technical reasons, character is used instead of word as the unit, making sure no word is cut at the extremities.

[2] This results in a total probability mass larger than 1; but still allows us to rank the probability of the senses.

[3] The logarithm was taken in base 10.

Word	WS	SW	Diff	Base	Bys	Ent
add.v	100	25	4.1	46	69	82
argument.n	175	75	3.1	47	45	54
ask.v	725	150	5.2	36	37	65
decide.v	725	375	5.2	77	65	75
different.a	175	0	4.0	47	34	48
eat.v	550	150	3.1	81	76	86
miss.v	425	125	5.1	28	40	53
simple.a	400	25	9.0	40	11	33
sort.n	175	75	4.0	66	60	71
use.v	50	25	5.6	58	57	79
wash.v	50	25	5.5	56	62	71

Table 1: Optimal window configuration and performance of both systems for the words on which Max Entropy has performed better than Naïve Bayes. (WS=Optimal window size; SW=Optimal sub-window size; Diff=Average absolute difference between the distribution of training and test samples; Accuracy (Base=Baseline; Bys=Naïve Bayes; Ent=Max Entropy)).

3. *Prep Bef, Prep Aft:* The last preposition before, and the first preposition after the target, if any.

4. *Sem Bef, Sem Aft:* The general semantic category of the noun before (after) the target. The category, which can be 'animate', 'inanimate', or 'abstract', is computed by traversing hypernym synsets of WordNet for all the senses of that noun. The first semantic category observed is returned, or 'inanimate' is returned as the default value.

The first three items are taken from Mihalcea's work (Mihalcea, 2002) which are useful features for most of the words. The range of all these features are closed sets; so Maximum Entropy is not biased by the *distribution of training samples among senses*, which is a side-effect of Naïve Bayes learners (see Section 5.2)[4].

The following is an example of the features extracted for sample *miss.v.bnc.00045286*: " ...? I'll *miss* the kids. But ...":

```
Article Bef=null,
POS Bef="MD", POS="VB", POS Aft="DT",
Prep Bef=null, Prep Aft=null,
Sem Bef=null, Sem After="animate"
```

---

[4]The Maximum Entropy program we used to learn these features was obtained from the OpenNLP site: http://maxent.sourceforge.net/index.html

Word Category	Naïve Bayes		Max Entropy	
	coarse	fine	coarse	fine
nouns	76%	70%	70%	61%
verbs	76%	67%	74%	66%
adjectives	59%	45%	59%	47%
Total	75%	67%	70%	63%

Table 2: Results of both approaches in fine and coarse grain evaluation.

5 Results and Discussion

The *Word Sense Disambiguator* program has been written as a *Processing Resource* in the Gate Architecture[5]. It uses the ANNIE *Tokenizer* and *POS Tagger* which are provided as components of Gate.

Table 2 shows the results of both systems for each category of words. It can be seen that approximate syntactic information has performed relatively better with adjectives which are generally harder to disambiguate.

5.1 Window size and the commonest effect

The optimal window size seems to be related to the distribution of the senses in the training samples and the number of training samples available for a word. Indeed, a large window size is selected when the number of samples is large, and the samples are not evenly distributed among senses. Basically because the words in Senseval are not mostly *topical* words, Naïve Bayes is working strongly with the commonest effect. On the other hand, when a small window size is selected, the commonest effect mostly vanishes and instead, collocations are relied upon.

5.2 Distribution of samples

A Naïve Bayes method is quite sensitive to the proportion of training and test samples: if the commonest class presented as test is different from the commonest class in training for example, this method performs poorly. This is a serious problem of Naïve Bayes towards real world WSD. For testing this claim, we made the following hypothesis: *When the mean of absolute difference of the test samples and training samples among classes of senses is more than 4%, Naïve Bayes method performs at most 20% above baseline*[6]. Table 3 shows that this hypothesis is confirmed in 82% of the cases (41 words

---

[5]http://www.gate.ac.uk/
[6]The following exceptional cases are not considered: 1) When baseline is above 70%, getting 20% above base-

	$Acc \leq 20$	$Acc > 20$
$Dist \leq 4.0$	5	26
$Dist > 4.0$	15	4

Table 3: Sensitivity of Naïve Bayes to the distribution of samples (Acc=Accuracy amount higher than baseline; Dist=Mean of distribution change.)

out of 50 ambiguous words that satisfy the conditions). Furthermore, such words are not necessarily difficult words. Our Maximum Entropy method performed on average 25% above the baseline on 7 of them (*ask.v, decide.v, different.a, difficulty.n, sort.n, use.v, wash.v* some of which are shown in Table 1).

5.3 Rare samples

Naïve Bayes mostly ignores the senses with a few samples in the training and gets its score on the senses with large number of training instances, while Maximum Entropy exploits features from senses which have had a few training samples.

5.4 Using lemmas and synsets

We tried working with word lemmas instead of derived forms; however, for some words it causes loss in accuracy. For example, for the adjective *different.a*, with window and sub-window size of 175 and 0, it reduces the accuracy from 60% to 46% with the validation set. However, for the noun *sort.n*, precision increases from 62% to 72% with a window size of 650 and sub-window size of 50. We believe that some senses come with a specific form of their neighboring tokens and lemmatization removes this distinguishing feature.

We also tried storing synsets of words as features for the Naïve Bayes learner, but obtained no significant change in the results.

6 Conclusion and Future Work

There is no fixed context window size applicable to all ambiguous words in the Naïve Bayes approach: keeping a large context window provides *domain information* which increases the resolution accuracy for some target words but not others. For non-topical words, large window size is selected only in order to exploit the distribution of samples.

Rough syntactic information performed well in our second system using Maximum Entropy modeling. This suggests that some senses can be strongly identified by syntax, leaving resolution of other senses to other methods. A simple, rough heuristic for recognizing when to rely on syntactic information in our system is when the selected window size by Naïve Bayes is relatively small.

We tried two simple methods for combining the two methods: considering context words as features in Max Entropy learner, and, establishing a separate Naïve Bayes learner for each syntactic/semantic feature and adding their scores to the basic contextual Naïve Bayes. These preliminary experiments did not result in any noticeable improvement.

Finally, using more semantic features from WordNet, such as *verb sub-categorization frames* (which are not consistently available) may help in distinguishing the senses.

Acknowledgments

References

N. Ide and J. Véronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40.

R. Mihalcea. 2002. Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proceedings of COLING'02*, Taiwan.

R. J. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of EMNLP-96*, pages 82–91, PA.

H. T. Ng. 1997. Exemplar-based word sense disambiguation: Some recent improvements. In *Proceedings of EMNLP-97*, pages 208–213. NJ.

E. Reifler. 1955. The mechanical determination of meaning. In *Machine translation of languages*, pages 136–164, New York. John Wiley and Sons.

---

line is really difficult, 2) When the difference is mostly on the commonest sense *being seen more than expected*, so the score is favored (7 words out of 57 satisfy these conditions.)