

Experiments in Statistical Word Sense Disambiguation: The CLaC System at Senseval-3

Abolfazl Keighobadi 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) at Senseval-3. We participated with 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, which puts this system in the middle of the ranking table among 37 supervised systems participating in the Senseval-3 english lexical task. 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 of this system.

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, a workshop held in cooperation with ACL 2004¹. 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 target word are selected as features to learn with a Maximum Entropy learner.

After showing the results of the systems, we will analyze the bias of the two methods to the data which results in a limit on the future features to be added to each system, and the effect of this bias in combination methods.

2 The Senseval-3 English lexical sample task

The Senseval competition is meant to sponsor research in word sense disambiguation (WSD) - that is, deciding which sense a word has in any given context. For example, in:

John has long arms, but short legs.

the word *arm* refers to a body part, while in:

John has a collection of military arms in his basements.

the word *arm* refers to guns.

Following Senseval-1 (1998) and Senseval-2 (2001), Senseval-3 took place in March-April 2004. The competition included 16 tasks ranging from the disambiguation of a set of pre-determined words, either a small set as in the *lexical sample task* (≈ 50 words) or a large set as in the *all words task* (≈ 5000 words) to finding semantic roles in the *automatic subcategorisation acquisition task*. The tasks involved various languages (English, Italian, Basque, Spanish, ...).

¹<http://www.senseval.org/senseval3>

Word Category	Training Data			Test Data
	Nb of target words	Nb of training samples for each target word	Average level of polysemy ²	Nb of testing samples for each target word
noun	20	180	5.35	90
verb	32	124	5.2	62
adjective	5	63	6.8	32
Average		138	5.8	69

Table 1: Description of the training and test data for the English lexical sample task

The training set We participated in the English lexical sample task, where a pre-defined set of 57 words had to be disambiguated. In total, 26 teams (47 systems) participated using both supervised and unsupervised methods. Each participant was given access to a training set of contexts with the target words annotated with their correct senses. The data, collected via the Open Mind Word Expert (OMWE) interface, used WordNet as sense inventory for nouns and adjectives, and Wordsmyth for verbs. The following is a sample of the training set for the noun *arm* with the sense `arm%1:08:00:: (body part)`.

```
<instance id="arm.n.bnc.00004403" docsrc="BNC">
<answer instance="arm.n.bnc.00004403" senseid="arm%1:08:00::"/>
<context>
Time will tell , sir , a colleague remarks , and Hawksmoor replies
: Time will not tell . Time never tells . Once more he raised his
<head>arm</head> involuntarily , as if in greeting . It is hard to think
that the novelist intended the reader to find this even more gnomie and
exasperating than the colleague seems to find it . But there may indeed be
some such aim .
</context>
</instance>
```

As Table 1 shows, the training data consisted of 57 words (20 nouns, 32 verbs and 5 adjectives). On average, the training set contained 138 context samples for each target word; however, the samples were not evenly distributed: more samples were available for each target noun than for verbs and adjectives. The average level of polysemy (the average number of possible senses for each target word) is more even among the different parts of speech; however, the senses are not evenly distributed within a specific target word as some senses are very rare while other are very frequent. These numbers seem to disadvantage adjectives as we have fewer training samples available, yet more senses to choose from.

Testing and Evaluation The task itself consisted of annotating the testing set by associating one or more senses with each occurrence of a target word along with the weight (or confidence level) for each sense. For example, the output lines:

```
brother.n 00001 501566
brother.n 00002 501566/0.5 501573/0.4 503751/0.1
```

indicate that the noun *brother* number 00001 is tagged as having the sense 501566 with a confidence level of 1; while the noun *brother* number 00002 is tagged as having the sense 501566 with a confidence level of 0.5, sense 501573 with a confidence level of 0.4 and sense 503751 with a confidence level of 0.1.

The test data contained a total of 184 instances to disambiguate. The distribution of these instances is given in the last column of Table 1.

²for the fine-grained evaluation

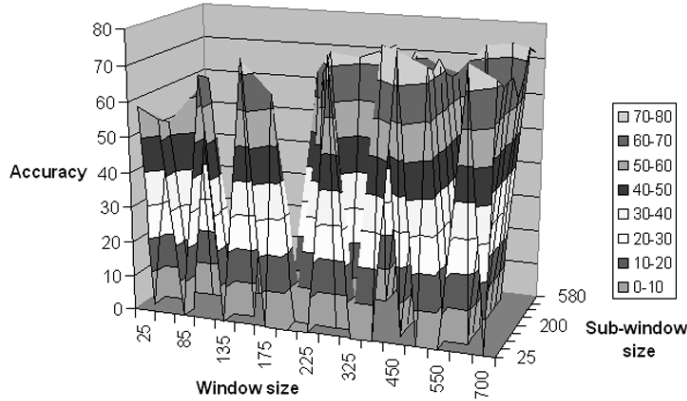


Figure 1: The effect of choosing different window and sub-window sizes for the word *bank.n*. The best accuracy is achieved with a window and sub-window size of around 450 and 50 characters respectively, while for example 50 and 25 provide very low accuracy.

The results were then evaluated using two types of granularity: coarse-grained and fine-grained, where the senses are organized into a subsumption hierarchy, given in a separate file: coarse-grained evaluation uses more general senses in the hierarchy, while fine-grained is based on more specific senses in the hierarchy. In both cases, for all target words of the test data, the sense labels given by the system is compared to the key labels and the match is weighted by the confidence level given. Three measures are then calculated:

precision: the number of correct labels (weighed) over the number of labeling attempts

recall: the number of correct labels (weighed) over the total number of target words

attempted: the ratio of attempted labeling over the total number of target words

Section 5 will present the result of our systems, but let us first describe the specifics of each system.

3 System 1: Naïve Bayes for Learning Context Words

The Naïve Bayes approach to WSD consist of taking a window of N words around the target word (the context), and computing the probability of their occurrences for each possible senses, using the probability distribution computed from the training set. More formally, during the testing phase, given the context of the target word, the score of every sense is computed using (Manning and Schütze, 1999):

$$Score_{sense_i} = \log p(sense_i) + \sum_k \log p(word_{k,i})$$

where, $p(word_{k,i})$ is the prior probability of word k inside the context window with respect to sense i .

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.

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

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 2: Optimal window configuration and performance of both systems for the words on which Maximum 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=Maximum Entropy)).

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 characters³. Table 2 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.

Approximate smoothing 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 words⁴. The intuition for our smoothing is that, with a small training set, more *unseen* words are likely to be encountered during the testing phase, and in order to prevent the accumulating penalty value from becoming relatively high, a small penalty for each word is selected. Additionally, the selection should not result in large differences in the computed scores of the senses. On the other hand, with a large training data, a stronger penalty is given to an unseen word (which are not encountered as frequently as the first case).

Finding proper probability (penalty) values for both cases was done experimentally; for a relatively large training data set⁵, $p(\text{an unseen word}) = 10^{-10}$ and for a small set, 10^{-9} resulted in the highest accuracy with our 15% validation set⁶.

4 System 2: Maximum Entropy learning of syntax and semantics

Another trend in WSD 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).

³For technical reasons, character is used instead of word as the unit, making sure no word is cut at the boundaries.

⁴This results in a total probability mass larger than 1, which needs to be normalized; but still allows us to rank the senses.

⁵When the total number of words seen during training is larger than 4300, or, the number of training instances is greater than 230, or, the selected context window size is larger than 400 characters.

⁶The logarithm was taken in base 10.

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.

Selected grammatical and semantic features 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 Before*: If there is any article before, the string token is considered as the value of this feature.
2. *POS, POS Before, POS After*: The part of speech of the target, the part of speech of the word before (after) if any.
3. *Prep Before, Prep After*: The last preposition before, and the first preposition after the target, if any.
4. *Sem Before, Sem After*: The general semantic category of the noun before (after) the target ambiguous word. This semantic category, which can be ‘animate’, ‘inanimate’, or ‘abstract’, is computed by traversing hypernym synsets for all the senses of that noun in WordNet. The first semantic category observed is returned, considering ‘inanimate’ 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 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"
```

To combine these features, we used a Maximum Entropy model⁷.

Maximum Entropy learning Maximum Entropy modeling originates from the hypothesis that faced with limited information, the most accurate representation is a model that makes the least assumptions about the future distribution of data. Consequently Maximum Entropy models assume a uniform distribution, only subject to constraints of prior evidence, i.e. the expectation that a combination of features produce a certain output. An elaborate explanation of Maximum Entropy theory as used in Natural Language Processing can be found in (Berger et al., 1996).

This model does not assume any specific distribution on the training data: we model all the facts that we know and assume nothing about the distribution of the facts that we do not know. To find the best combination of features, in effect, among all the models consistent with the known facts, we select the model with the greatest entropy - the model that maximizes the uncertainty measure.

Because the range of all the features (POS and the semantic tags) are closed sets, *it is less likely to introduce unique feature combinations*; for this reason we chose different features than context words.

As we will see in section 6, this method is biased towards *uniqueness of a combination of features*. Because of implementation issues, we didn’t explore weighted features and smoothing in Maximum Entropy.

⁷The Maximum Entropy program we used to combine these features was obtained from the OpenNLP site: <http://maxent.sourceforge.net/index.html>

Word Category	Naïve Bayes		Maximum 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 3: Results of both approaches in fine and coarse-grained evaluation.

	coarse	fine
Baseline (most frequent sense)	64.5%	55.2%
Best system	79.5%	72.9%

Table 4: Official results of the baseline and best system for English lexical task.

5 Results and Discussion

The *Word Sense Disambiguator* program has been written as a *Processing Resource* in the Gate Architecture⁸. It uses the *ANNIE Tokenizer* and *POS Tagger* which are provided as components of Gate.

Table 3 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 (see section 2). Table 4 shows the official best system and the baseline results: both our methods performed better than the baseline (always picking the most common sense). Our system 1 (Naïve Bayes) is the median in the ranking but system 2 (Maximum Entropy) was lower than the median (Mihalcea et al., 2004).

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.

Rare samples Because of high effect of the prior probability of seeing each sense, 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.

Using lemmas and synsets We tried working with word lemmas instead of derivated 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 Learning Bias in the Two Systems

As can be observed from the analysis of the results, the difference between the two learning methods employed in our experiments goes beyond performance. While the level of performance of the Naïve Bayes method was generally higher, specific words and senses favored the Maximum Entropy approach (refer to Table 2). There is sufficient evidence to attribute the explanation to the phe-

⁸<http://www.gate.ac.uk/>

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

Table 5: Sensitivity of Naïve Bayes to the distribution of samples (Acc=Accuracy amount higher than baseline; Dist=Mean of distribution change.)

nomenon of bias. According to Mooney (Mooney, 1996), a specific learning method is favorable for a specific class of problems due to bias, or the method’s basis of generalization.

Mooney discussed the effect of bias on inductive learning methods; for example the Naïve Bayes method is biased to the independence of the features.

Naïve Bayes and the distribution of samples The Naïve Bayes method is quite sensitive to the proportion of training versus test samples: if the commonest class presented as test is different from the commonest class in training for example, this method performs poorly. Additionally, “*Naïve Bayes heavily favors classes with more training examples*” (Rennie et al., 2003). This is a serious problem of Naive Bayes towards real world WSD.

Naïve Bayes assumes that the features are independent of each other. With this assumption, prior probabilities are computed for every feature. These probabilities however work only if the same distribution of samples are seen during testing (i.e. if we have a smooth world which has been modeled by seeing enough instances - which we will see is not possible). So, when features are not independent, or when the distribution of samples is not the same in testing data, this method will not perform well.

The first case, where features are not independent is mathematically obvious: if features are dependent on each other then

$$p(\text{context}|\text{sense}_i) \neq \prod_k p(\text{word}_k|\text{sense}_i)$$

For the effect of different distributions, we made the following observation based on the results of the training test: *When the mean of absolute difference of the distribution of the test samples and training samples among classes of senses is more than 4%, Naïve Bayes method performs at most 20% above baseline*⁹. Table 5 shows that this result is confirmed in 82% of the cases (41 words 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 the 15 words with high change in their distribution (*ask.v*, *decide.v*, *different.a*, *difficulty.n*, *sort.n*, *use.v*, *wash.v* some of which are shown in Table 2).

Bias in Maximum Entropy In Maximum Entropy, bias is for uniform distribution of samples that satisfy constraints as seen in training. (Fleischman et al., 2003) found that semantic role classification benefits from this bias. In WSD, this translates to a bias for combinations of features appearing more often with a particular sense. Another significant difference with the Naïve Bayes approach is that features are not perceived as independent. Each single feature is considered in its different combinations with other features.

Combination of contextual and syntactic/semantic features As it was discussed above, Naïve Bayes is based on the independence assumption of the features; so features which are naturally dependent (syntactic/semantic features) can not be learned with this method. This is why we used a different learner for these features.

7 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.

⁹The following exceptional cases are not considered in testing this hypothesis: 1) When baseline is above 70%, getting 20% above baseline 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.)

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.

The problem of WSD for each particular word could be seen as a different classification problem. Single word classifiers based on this approach could achieve very high accuracy levels, however, due to limited resources, it is impractical to build a classifier for each word in a language. Our experiments show that a Naïve Bayes and a Maximum Entropy classifiers are suitable for different words and senses.

We tried two simple ways for combining the two methods: considering context words as features in Maximum 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.

Approaches towards a combination would be either a learner on top of the two systems, as well as based on a linguistic investigation of the ambiguous words to group them in specific linguistic categories (such as grammatical, topical, etc.). We also intend to experiment with adding other types of classifiers in cases where both the above methods perform poorly.

Using more semantic features from WordNet, such as *verb sub-categorization frames* (which are not consistently available) may help in distinguishing the senses with Maximum Entropy. On the other hand, combining the grammatical features in one feature (in the form of a vector) may provide Naïve Bayes with dependent features to be learned properly.

Finally, we believe that Senseval data is not representative of real world WSD problems, since the distribution of samples is artificially prepared to match that of the training samples. The same distribution does not necessary occur in different genres.

Acknowledgments

Many thanks to Glenda B. Anaya and Michelle Khalifé for their invaluable help.

References

- Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.
- M. Fleischman, Namhee Kwon, and E. Hovy. 2003. Maximum entropy models for framenet classification. In *Empirical Methods in Natural Language Processing*, Sapporo, Japan.
- 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.
- C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.
- R. Mihalcea, T. Chklovski, and A. Kilgarriff. 2004. The senseval-3 english lexical sample task. In *Proceedings of Senseval-3*, pages 25–28, Barcelona, Spain.
- 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.
- Jason D. Rennie, Lawrence Shih, Jaime Teevan, and David Karger. 2003. Tackling the poor assumptions of naive bayes text classifiers. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, pages 616–623, USA.