Naive Bayes and Exemplar-Based approaches to Word Sense Disambiguation Revisited

Gerard Escudero, Lluís Màrquez and **German Rigau**[1]

Abstract. This paper describes an experimental comparison between two standard supervised learning methods, namely Naive Bayes and Exemplar–based classification, on the Word Sense Disambiguation (WSD) problem. The aim of the work is twofold. Firstly, it attempts to contribute to clarify some confusing information about the comparison between both methods appearing in the related literature. In doing so, several directions have been explored, including: testing several modifications of the basic learning algorithms and varying the feature space. Secondly, an improvement of both algorithms is proposed, in order to deal with large attribute sets. This modification, which basically consists in using only the *positive* information appearing in the examples, allows to improve greatly the efficiency of the methods, with no loss in accuracy. The experiments have been performed on the largest sense–tagged corpus available containing the most frequent and ambiguous English words. Results show that the Exemplar–based approach to WSD is generally superior to the Bayesian approach, especially when a specific metric for dealing with symbolic attributes is used.

1 INTRODUCTION

Word Sense Disambiguation (WSD) is the problem of assigning the appropriate meaning (sense) to a given word in a text or discourse. Resolving the ambiguity of words is a central problem for language understanding applications and their associated tasks [7], including, for instance, machine translation, information retrieval and hypertext navigation, parsing, speech synthesis, spelling correction, reference resolution, automatic text summarization, etc.

WSD is one of the most important open problems in the Natural Language Processing (NLP) field. Despite the wide range of approaches investigated and the large effort devoted to tackle this problem, it is a fact that to date no large–scale broad–coverage and highly accurate word sense disambiguation system has been built.

One of the most successful current lines of research is the corpus–based approach in which statistical or Machine Learning (ML) algorithms have been applied to learn statistical models or classifiers from corpora in order to perform WSD. Generally, supervised approaches (those that learn from a previously semantically annotated corpus) have obtained better results than unsupervised methods on small sets of selected highly ambiguous words, or artificial pseudo–words. Many standard ML algorithms for supervised learning have been applied, such as: Bayesian learning [16, 19], Exemplar–based learning [18, 16, 5], Decision Lists [21], Neural Networks [20], etc. Further, Mooney [15] provides a comparative experiment on a very restricted domain between all previously cited methods but also including Decision Trees and Rule Induction algorithms.

Despite the good results obtained on limited domains, supervised methods suffer from the lack of widely available semantically tagged corpora, from which to construct really broad coverage systems. This is known as the "knowledge acquisition bottleneck" [6]. Ng [17] estimates that the manual annotation effort necessary to build a broad coverage semantically annotated corpus would be about 16 man-years. This extremely high overhead for supervision and, additionally, the also serious learning overhead when common ML algorithms scale to real size WSD problems, explain why supervised methods have been seriously questioned.

Due to this fact, recent works have focused on reducing the acquisition cost as well as the need for supervision of corpus–based methods for WSD. Consequently, the following three lines of research are currently being studied: 1) The design of efficient example sampling methods [4, 5]; 2) The use of lexical resources, such as WordNet [13], and WWW search engines to automatically obtain from Internet accurate and arbitrarily large word sense samples [8, 12]; 3) The use of unsupervised EM–like algorithms for estimating the statistical model parameters [19]. It is our belief that this body of work, and in particular the second line, provide enough evidence towards the "opening" of the acquisition bottleneck in the near future. For that reason, it is worth further investigating the application of supervised ML methods to WSD, and thoroughly comparing existing alternatives.

1.1 Comments about Related Work

Unfortunately, there have been very few direct comparisons between alternative methods for WSD. However, it is commonly stated that Naive Bayes, Neural Networks and Exemplar–based learning represent state–of–the–art accuracy on supervised WSD [15, 16, 8, 5, 19]. Regarding the comparison between Naive Bayes and Exemplar–based methods, the works by Mooney [15] and Ng [16] will be the ones basically referred to in this paper.

Mooney's paper shows that the Bayesian approach is clearly superior to the Exemplar–based approach. Although it is not explicitly said, the overall accuracy of Naive Bayes is about 16 points higher than that of the Example–based algorithm, and the latter is only slightly above the accuracy that a Most–Frequent–Sense classifier would obtain. In the Exemplar–based approach, the algorithm applied for classifying new examples was a standard k-Nearest–Neighbour (k-NN), using the Hamming distance for measuring closeness. Neither example weighting nor attribute weighting are applied, k is set to 3, and the number of attributes used is said to be almost 3,000.

The second paper compares the Naive Bayes approach with PE-

[1] TALP Research Center, Software Department, Technical University of Catalonia, Jordi Girona Salgado 1-3, Barcelona E-08034, Catalonia, email: {escudero, lluism, g.rigau}@lsi.upc.es

BLS [1], a more sophisticated Exemplar–based learner especially designed for dealing with examples that have symbolic features. This paper shows that, for a large number of nearest–neighbours, the performance of both algorithms is comparable, while if cross validation is used for parameter setting, PEBLS slightly outperforms Naive Bayes. It has to be noted that the comparison was carried out in a limited setting, using only 7 features, and that the attribute/example–weighting facilities provided by PEBLS were not used. The author suggests that the poor results obtained in Mooney's work were due to the metric associated to the k-NN algorithm, but he did not test if the MVDM metric used in PEBLS is superior to the standard Hamming distance or not.

Another surprising result that appears in Ng's paper is that the accuracy results obtained were 1–1.6% higher than those reported by the same author one year before [18], when running exactly the same algorithm on the same data, but using a larger and richer set of attributes. This apparently paradoxical difference is attributed, by the author, to the feature pruning process performed in the older paper.

Apart from the contradictory results obtained by the previous papers, some methodological drawbacks of both comparisons should also be pointed out. On the one hand, Ng applies the algorithms on a broad–coverage corpus but reports the accuracy results of a single testing experiment, providing no statistical tests of significance. On the other hand, Mooney performs thorough and rigorous experiments, but he compares the alternative methods on a limited domain consisting of a single word with a reduced set of six senses. Thus, it is our claim that this extremely specific domain does not guarantee the reaching of reliable conclusions about the relative performances of alternative methods when applied to broad–coverage domains.

Consequently, the aim of this paper is twofold: 1) To study the source of the differences between both approaches in order to clarify the contradictory and incomplete information. 2) To empirically test the alternative algorithms and their extensions on a broad–coverage sense tagged corpus, in order to estimate which is the most appropriate choice.

The paper is organized as follows: Section 2 describes the algorithms that will be tested, as well as the notation used. Section 3 is devoted to carefully explain the experimental setting. Section 4 reports the set of experiments performed and the analysis of the results obtained. The best alternative methods are tested on a broad coverage corpus in Section 5. Finally, Section 6 concludes and outlines some directions for future work.

2 BASIC METHODS

2.1 Naive Bayes

The Naive Bayes classifier has been used in its most classical setting [3]. Let $C_1 \ldots C_m$ the different classes and $\bigcap v_j$ the set of feature values of a test example. The Naive Bayes method tries to find the class that maximizes $P(C_i \mid \cap v_j)$. Assuming independence between features, the goal of the algorithm can be stated as:

$$\arg\max_i \; P(C_i \mid \cap v_j) \approx \arg\max_i \; P(C_i) \prod_j P(v_j \mid C_i)\,,$$

where $P(C_i)$ and $P(v_j \mid C_i)$ are estimated during training process using relative frequencies. To avoid the effects of zero counts when estimating the conditional probabilities of the model, a very simple smoothing technique, proposed in Ng's paper [16], has been used. It consists in replacing zero counts of $P(v_j \mid C_i)$ with $P(C_i)/N$ where N is the number of training examples.

Hereinafter, this method will be referred to as NB.

2.2 Exemplar-Based Approach

In our basic implementation all examples are stored in memory and the classification of a new example is based on a k–NN algorithm, which uses Hamming distance to measure closeness (in doing so, all examples are examined). If k is greater than 1, the resulting sense is the majority sense of the k nearest neighbours. Ties are resolved in favour of the most frequent sense among all those tied. Hereinafter, this algorithm will be referred to as $\mathsf{EB}_{h,k}$.

In order to test some of the hypotheses about the differences between Naive Bayes and Exemplar–based approaches, some variants of the basic k-NN algorithm have been implemented:

- **Example weighting**. This variant introduces a simple modification in the voting scheme of the k nearest neighbours, which makes the contribution of each example proportional to their importance. When classifying a new test example, each example of the set of nearest neighbours votes for its class with a weight proportional to its closeness to the test example. Hereinafter, this variant will be referred to as $\mathsf{EB}_{h,k,e}$.
- **Attribute weighting**. This variant consists of ranking all attributes by relevance and making them contribute to the distance calculation with a weight proportional to their importance. The attribute weighting has been done using the RLM distance measure [9]. This measure, belonging to the distance/information–based families of attribute selection functions, has been selected because it showed better performance than seven other alternatives in an experiment of decision tree induction for PoS tagging [11]. Hereinafter, this variant will be referred to as $\mathsf{EB}_{h,k,a}$.

When both modifications are put together, the resulting algorithm will be referred to as $\mathsf{EB}_{h,k,e,a}$. Finally, we have also investigated the effect of using an alternative metric.

- **Modified Value Difference Metric** (MVDM), proposed by Cost and Salzberg [1], allows making graded guesses of the match between two different symbolic values. Let v_1 and v_2 be two values of a given attribute a. The MVDM distance between them is defined as:

$$d(v_1, v_2) = \sum_{i=1}^{m} |P(C_i|v_1) - P(C_i|v_2)| \approx \sum_{i=1}^{m} \left| \frac{N_{1,i}}{N_1} - \frac{N_{2,i}}{N_2} \right|$$

  where m is the number of classes, $N_{x,i}$ is the number of training examples with value v_x of attribute a that are classified as class i in the training corpus and N_x is the number of training examples with value v_x of attribute a in any class. Hereinafter, this variant will be referred to as $\mathsf{EB}_{cs,k}$. This algorithm has also been used with the example–weighting facility ($\mathsf{EB}_{cs,k,e}$).

3 SETTING

In our experiments, both approaches have been evaluated on the DSO corpus, a semantically annotated corpus containing 192,800 occurrences of 121 nouns and 70 verbs[2], corresponding to the most frequent and ambiguous English words. This corpus was collected by Ng and colleagues [18] and it is available from the Linguistic Data Consortium (LDC)[3].

[2] These examples, consisting of the full sentence in which the ambiguous word appears, are tagged with a set of labels corresponding, with minor changes, to the senses of WordNet 1.5 [13].

[3] LDC address: `http://www.ldc.upenn.edu/`

For our first experiments, a group of 15 words (10 nouns and 5 verbs) which frequently appear in the WSD literature has been selected. These words are described in the left hand–side of table 1. Since our goal is to acquire a classifier for each word, each row represents a classification problem. The number of classes (senses) ranges from 4 to 30 and the number of training examples ranges from 373 to 1,500. The MFS column of the table 1 show the percentage of the most frequent sense for each word, i.e. the accuracy that a naive "Most–Frequent–Sense" classifier would obtain.

Table 1. Set of 15 reference words.

				%	# Attributes	
Word	POS	Sens.	Exs.	MFS	SETA	SETB
age	n	4	493	62.1	7	3,015
art	n	5	405	46.7	7	2,641
car	n	5	1,381	95.1	7	4,719
child	n	4	1,068	80.9	7	4,840
church	n	4	373	63.1	7	2,375
cost	n	3	1,500	87.3	7	4,930
fall	v	19	1,500	70.1	7	4,173
head	n	14	870	36.9	7	4,284
interest	n	7	1,500	45.1	7	5,328
know	v	8	1,500	34.9	7	5,301
line	n	26	1,342	21.9	7	5,813
set	v	19	1,311	36.9	7	5,749
speak	v	5	517	69.1	7	2,975
take	v	30	1,500	35.6	7	6,428
work	n	7	1,469	31.7	7	6,321
Avg. nouns		8.6	1,040.1	57.4	7	4,935.0
verbs		17.9	1,265.6	46.6	7	5,203.5
all		12.1	1,115.3	53.3	7	5,036.6

Two sets of attributes have been used, which will be referred to as SETA and SETB, respectively. Let "... w_{-3} w_{-2} w_{-1} w w_{+1} w_{+2} w_{+3} ..." be the context of consecutive words around the word w to be disambiguated. Attributes refer to this context as follows.

- SETA contains the seven following attributes: w_{-2}, w_{-1}, w_{+1}, w_{+2}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1}), and (w_{+1}, w_{+2}), where the last three correspond to collocations of two consecutive words. These attributes, which are exactly those used in [16], represent the *local context* of the ambiguous word and they have been proven to be very informative features for WSD. Note that whenever an attribute refers to a position that falls beyond the boundaries of the sentence for a certain example, a default value "_" is assigned.

Let $p_{\pm i}$ be the part–of–speech tag of word $w_{\pm i}$, and $c_1, \ldots, c_m$ the unordered set of open class words appearing in the sentence.

- SETB enriches the local context: w_{-1}, w_{+1}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1}), (w_{+1}, w_{+2}), (w_{-3}, w_{-2}, w_{-1}), (w_{-2}, w_{-1}, w_{+1}), (w_{-1}, w_{+1}, w_{+2}) and (w_{+1}, w_{+2}, w_{+3}), with the part–of–speech information: p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, p_{+3}, and, additionally, it incorporates *broad context* information: $c_1 \ldots c_m$. SETB is intended to represent a more realistic set of attributes for WSD[4]. Note that c_i attributes are binary–valued, denoting the the presence or absence of a content word in the sentence context.

The right hand–side of table 1 contains the information about the number of features. Note that SETA has a constant number of attributes (7), while for SETB this number depends on the concrete word, and that it ranges from 2,641 to 6,428.

[4] In fact, it incorporates all the attributes used in [18], with the exception of the morphology of the target word and the verb–object syntactic relation.

4 EXPERIMENTS

The comparison of algorithms has been performed in series of controlled experiments using exactly the same training and test sets for each method. The experimental methodology consisted on a 10-fold cross-validation. All accuracy/error rate figures appearing in the paper are averaged over the results of the 10 folds. The statistical tests of significance have been performed using a 10-fold cross validation paired Student's t-test [2] with a confidence value of: $t_{9,0.975} = 2.262$.

Exemplar-based algorithms are run several times using different number of nearest neighbours (1, 3, 5, 7, 10, 15, 20 and 25) and the results corresponding to the best choice are reported[5].

4.1 Using SETA

Table 2 shows the results of all methods and variants tested on the 15 reference words, using the SETA set of attributes: Most Frequent Sense (MFS), Naive Bayes (NB), Exemplar–based using Hamming distance (EB_h variants, 5th to 9th columns), and Exemplar-based approach using the MVDM metric (EB_{cs} variants, 10th to 12th columns) are included. The best result for each word is printed in boldface. From these figures, several conclusions can be drawn:

- All methods significantly outperform the MFS classifier.
- Referring to the EB_h variants, $\text{EB}_{h,7}$ performs significantly better than $\text{EB}_{h,1}$, confirming the results of Ng [16] that values of k greater than one are needed in order to achieve good performance with the k-NN approach. Additionally, both example weighting ($\text{EB}_{h,15,e}$) and attribute weighting ($\text{EB}_{h,7,a}$) significantly improve $\text{EB}_{h,7}$. Further, the combination of both ($\text{EB}_{h,7,e,a}$) achieves an additional improvement.
- The MVDM metric is superior to Hamming distance. The accuracy of $\text{EB}_{cs,10,e}$ is significantly higher than those of any EB_h variant. Unfortunately, the use of weighted examples does not lead to further improvement in this case. A drawback of using the MVDM metric is the computational overhead introduced by its calculation. Table 4 shows that EB_h is fifty times faster than EB_{cs} using SETA[6].
- The Exemplar-based approach achieves better results than the Naive Bayes algorithm. This difference is statistically significant when comparing the $\text{EB}_{cs,10}$ and $\text{EB}_{cs,10,e}$ against NB.

4.2 Using SETB

The aim of the experiments with SETB is to test both methods with a realistic large set of features. Table 3 summarizes the results of these experiments[7].

Let's now consider only NB and EB_h (3rd and 5th columns). A very surprising result is observed: while NB achieves almost the same accuracy that in the previous experiment, the exemplar–based approach shows a very low performance. The accuracy of EB_h drops 8.6 points (from 6th column of table 2 to 5th column of table 3) and is only slightly higher than that of MFS.

[5] In order to construct a real k-NN–based system for WSD, the k parameter should be estimated by cross–validation using only the training set [16], however, in our case, this cross–validation inside the cross–validation involved in the testing process would generate a prohibitive overhead.

[6] The current programs are implemented using PERL-5.003 and they run on a SUN UltraSPARC-2 machine with 192Mb of RAM.

[7] Detailed results for each word are not included.

Table 2. Results of all algorithms on the set of 15 reference words using SETA.

Word	POS	Accuracy (%)									
		MFS	NB	$\mathsf{EB}_{h,1}$	$\mathsf{EB}_{h,7}$	$\mathsf{EB}_{h,15,e}$	$\mathsf{EB}_{h,7,a}$	$\mathsf{EB}_{h,7,e,a}$	$\mathsf{EB}_{cs,1}$	$\mathsf{EB}_{cs,10}$	$\mathsf{EB}_{cs,10,e}$
age	n	62.1	73.8	71.4	69.4	71.0	74.4	**75.9**	70.8	73.6	73.6
art	n	46.7	54.8	44.2	59.3	58.3	58.5	57.0	54.1	59.5	**61.0**
car	n	95.1	95.4	91.3	95.5	95.8	96.3	96.2	95.4	**96.8**	**96.8**
child	n	80.9	86.8	82.3	89.3	89.5	91.0	**91.2**	87.5	91.0	90.9
church	n	61.1	62.7	61.9	62.7	63.0	62.5	64.1	61.7	**64.6**	64.3
cost	n	87.3	86.7	81.1	87.9	87.7	**88.1**	87.8	82.5	85.4	84.7
fall	v	70.1	76.5	73.3	78.2	79.0	78.1	79.8	78.7	81.6	**81.9**
head	n	36.9	76.9	70.0	76.5	76.9	77.0	78.7	74.3	78.6	**79.1**
interest	n	45.1	64.5	58.3	62.4	63.3	64.8	66.1	65.1	67.3	**67.4**
know	v	34.9	47.3	42.2	44.3	46.7	44.9	46.8	45.1	49.7	**50.1**
line	n	21.9	51.9	46.1	47.1	49.7	50.7	51.9	53.3	**57.0**	56.9
set	v	36.9	55.8	43.9	53.0	54.8	52.3	54.3	49.7	**56.2**	56.0
speak	v	69.1	**74.3**	64.6	72.2	73.7	71.8	72.9	67.1	72.5	72.9
take	v	35.6	44.8	39.3	43.7	46.1	44.5	46.0	45.3	48.8	**49.1**
work	n	31.7	51.9	42.5	43.7	47.2	48.5	48.9	48.5	52.0	**52.5**
Avg. nouns		57.4	71.7	65.8	70.0	71.1	72.1	72.6	70.6	73.6	**73.7**
verbs		46.6	57.6	51.1	56.3	58.1	56.4	58.1	55.9	60.3	**60.5**
all		53.3	66.4	60.2	64.8	66.2	66.1	67.2	65.0	68.6	**68.7**

Table 3. Results of all algorithms on the set of 15 reference words using SETB.

POS	Accuracy (%)											
	MFS	NB	PNB	$\mathsf{EB}_{h,15}$	$\mathsf{PEB}_{h,1}$	$\mathsf{PEB}_{h,7}$	$\mathsf{PEB}_{h,7,e}$	$\mathsf{PEB}_{h,7,a}$	$\mathsf{PEB}_{h,10,e,a}$	$\mathsf{PEB}_{cs,1}$	$\mathsf{PEB}_{cs,10}$	$\mathsf{PEB}_{cs,10,e}$
nouns	57.4	72.2	72.4	64.3	70.6	72.4	73.7	72.5	73.4	73.2	75.4	**75.6**
verbs	46.6	55.2	55.3	43.0	54.7	57.7	59.5	58.9	60.2	58.6	61.9	**62.1**
all	53.3	65.8	66.0	56.2	64.6	66.8	**68.4**	67.4	68.4	67.7	70.3	**70.5**

The problem is that the binary representation of the broad–context attributes is not appropriate for the k-NN algorithm. Such a representation leads to an extremely sparse vector representation of the examples, since in each example only a few words, among all possible, are observed. Thus, the examples are represented by a vector of about 5,000 0's and only a few 1's. In this situation two examples will coincide in the majority of the values of the attributes (roughly speaking in "all" the zeros) and will probably differ in those positions corresponding to 1's. This fact wrongly biases the similarity measure (and thus the classification) in favour of that stored examples which have less 1's, that is, those corresponding to the shortest sentences.

This situation could explain the poor results obtained by the k-NN algorithm in Mooney's work, in which a large number of attributes was used. Further, it could explain why the results of Ng's system working with a rich attribute set (including binary–valued contextual features) were lower than those obtained with a simpler set of attributes[8].

In order to address this limitation we propose to reduce the attribute space by collapsing all binary attributes $c_1, \ldots, c_m$ in a single set–valued attribute c that contains, for each example, all content words that appear in the sentence. In this setting, the similarity S between two values $V_i = \{w_{i_1}, w_{i_2}, \ldots, w_{i_n}\}$ and $V_j = \{w_{j_1}, w_{j_2}, \ldots, w_{j_m}\}$ can be redefined as: $S(V_i, V_j) = \parallel V_i \cap V_j \parallel$, that is, equal to the number of words shared[9].

This approach implies that a test example is classified taking into account the information about the words it contains (*positive* information), but no the information about the words it does not contain. Besides, it allows a very efficient implementation, which will be referred to as PEB (standing for Positive Exemplar–Based).

In the same direction, we have tested the Naive Bayes algorithm combining only the conditional probabilities corresponding to the words that appear in the test examples. This variant is referred to as PNB. The results of both PEB and PNB are included in table 3, from which the following conclusions can be drawn.

- The PEB approach reaches excellent results, improving by 10.6 points the accuracy of EB (see 5th and 7th columns of table 3). Further, the results obtained significantly outperform those obtained using SETA, indicating that the (careful) addition of richer attributes leads to more accurate classifiers. Additionally, the behaviour of the different variants is similar to that observed when using SETA, with the exception that the addition of attribute–weighting to the example–weighting ($\mathsf{PEB}_{h,10,e,a}$) seems no longer useful.
- PNB algorithm is at least as accurate as NB.
- Table 4 shows that the *positive* approach increases greatly the efficiency of the algorithms. The acceleration factor is 80 for NB and 15 for EB_h (the calculation of EB_{cs} variants was simply not feasible working with the attributes of SETB).
- The comparative conclusions between the Bayesian and Exemplar–based approaches reached in the experiments using SETA also hold here. Further, the accuracy of $\mathsf{PEB}_{h,7,e}$ is now significantly higher than that of PNB.

Table 4. CPU–time elapsed on the set of 15 words ("hh:mm").

	NB	$\mathsf{EB}_{h,15,e}$	$\mathsf{EB}_{h,7,a}$	$\mathsf{EB}_{cs,10,e}$
SETA	00:07	00:08	00:11	09:56

	NB	PNB	$\mathsf{EB}_{h,15,e}$	$\mathsf{PEB}_{h,7,e}$	$\mathsf{PEB}_{h,7,a}$	$\mathsf{PEB}_{cs,10,e}$
SETB	16:13	00:12	06:04	00:25	03:55	49:43

5 GLOBAL RESULTS

In order to ensure that the results obtained so far also hold on a realistic broad–coverage domain, the PNB and PEB algorithms have

[8] Recall that authors attributed the bad results to the absence of attribute weighting and to the attribute pruning, respectively.

[9] This measure is usually known as the *matching coefficient* [10]. More complex similarity measures, e.g. Jaccard or Dice coefficients, have not been explored.

been tested on the whole sense–tagged corpus, using both sets of attributes. This corpus contains about 192,800 examples of 121 nouns and 70 verbs. The average number of senses is 7.2 for nouns, 12.6 for verbs, and 9.2 overall. The average number of training examples is 933.9 for nouns, 938.7 for verbs, and 935.6 overall.

The results obtained are presented in table 5. It has to be noted that the results of PEB_{cs} using SETB were not calculated due to the extremely large computational effort required by the algorithm (see table 4). Results are coherent to those reported previously, that is:

Table 5. Global results on the 191–word corpus.

		Accuracy (%)				CPU–Time (hh:mm)		
	POS	MFS	PNB	PEB_h	PEB_{cs}	PNB	PEB_h	PEB_{cs}
SETA	nouns	56.4	68.7	68.5	**70.2**	00:33	00:47	92:22
	verbs	48.7	64.8	65.3	**66.4**			
	all	53.2	67.1	67.2	**68.6**			
SETB	nouns	56.4	69.2	**70.1**	–	01:06	01:46	–
	verbs	48.7	63.4	**67.0**				
	all	53.2	66.8	**68.8**				

- In SETA, the Exemplar–based approach using the MVDM metric is significantly superior to the rest.
- In SETB, the Exemplar–based approach using Hamming distance and example weighting significantly outperforms the Bayesian approach. Although the use of the MVDM metric could lead to better results, the current implementation is computationally prohibitive.
- Contrary to the Exemplar-based approach, Naive Bayes does not improve accuracy when moving from SETA to SETB, that is, the simple addition of attributes does not guarantee accuracy improvements in the Bayesian framework.

6 CONCLUSIONS

This work has focused on clarifying some contradictory results obtained when comparing Naive Bayes and Exemplar–based approaches to WSD. Different alternative algorithms have been tested using two different attribute sets on a large sense–tagged corpus. The experiments carried out show that Exemplar–based algorithms have generally better performance than Naive Bayes, when they are extended with example/attribute weighting, richer metrics, etc.

The reported experiments also show that the Exemplar–based approach is very sensitive to the representation of a concrete type of attributes, frequently used in Natural Language problems. To avoid this drawback, an alternative representation of the attributes has been proposed and successfully tested. Furthermore, this representation also improves the efficiency of the algorithms, when using a large set of attributes.

The test on the whole corpus allows us to estimate that, in a realistic scenario, the best tradeoff between performance and computational requirements is achieved by using the Positive Exemplar–based algorithm, SETB set of attributes, Hamming distance, and example–weighting.

Further research on the presented algorithms to be carried out in the near future includes: 1) The study of the behaviour with respect to the number of training examples; 2) The study of the robustness in the presence of highly redundant attributes; 3) The testing of the algorithms on alternative sense–tagged corpora automatically acquired from Internet.

ACKNOWLEDGEMENTS

This research has been partially funded by the Spanish Research Department (CICYT's project TIC98–0423–C06) and by the Catalan Research Department (CIRIT's consolidated research group 1999SGR-150, CREL's Catalan WordNet project and CIRIT's grant 1999FI 00773).

We would also like to thank the referees for their valuable comments.

REFERENCES

[1] S. Cost and S. Salzberg, 'A weighted nearest neighbor algorithm for learning with symbolic features', *Machine Learning*, **10**(1), 57–78, (1993).

[2] T. G. Dietterich, 'Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms', *Neural Computation*, **10**(7), (1998).

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[4] S. P. Engelson and I. Dagan, 'Minimizing Manual Annotation Cost in Supervised Training from Corpora', in *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, eds., E. Riloff S. Wermter and G. Scheler, LNAI, 1040, Springer, (1996).

[5] A. Fujii, K. Inui, T. Tokunaga, and H. Tanaka, 'Selective Sampling for Example–based Word Sense Disambiguation', *Computational Linguistics*, **24**(4), 573–598, (1998).

[6] W. Gale, K. W. Church, and D. Yarowsky, 'A Method for Disambiguating Word Senses in a Large Corpus', *Computers and the Humanities*, **26**, 415–439, (1993).

[7] N. Ide and J. Véronis, 'Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art', *Computational Linguistics*, **24**(1), 1–40, (1998).

[8] C. Leacock, M. Chodorow, and G. A. Miller, 'Using Corpus Statistics and WordNet Relations for Sense Identification', *Computational Linguistics*, **24**(1), 147–166, (1998).

[9] R. Lopez de Mántaras, 'A Distance–Based Attribute Selection Measure for Decision Tree Induction', *Machine Learning*, **6**(1), 81–92, (1991).

[10] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, 1999.

[11] L. Màrquez, *Part–of–Speech Tagging: A Machine–Learning Approach based on Decision Trees*, Phd. Thesis, Software Department, Technical University of Catalonia, 1999.

[12] R. Mihalcea and I. Moldovan, 'An Automatic Method for Generating Sense Tagged Corpora', in *Proceedings of the 16th National Conference on Artificial Intelligence*. AAAI Press, (1999).

[13] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, 'Five Papers on WordNet', *Special Issue of International Journal of Lexicography*, **3**(4), (1990).

[14] G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker, 'A Semantic Concordance', in *Proceedings of the ARPA Workshop on Human Language Technology*, (1993).

[15] R. J. Mooney, 'Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning', in *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*, (1996).

[16] H. T. Ng, 'Exemplar-Base Word Sense Disambiguation: Some Recent Improvements', in *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*, (1997).

[17] H. T. Ng, 'Getting Serious about Word Sense Disambiguation', in *Proceedings of the ACL SIGLEX Workshop "Tagging Text with Lexical Semantics: Why, what and how?"*, (1997).

[18] H. T. Ng and H. B. Lee, 'Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach', in *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. ACL, (1996).

[19] T. Pedersen and R. Bruce, 'Knowledge Lean Word-Sense Disambiguation', in *Proceedings of the 15th National Conference on Artificial Intelligence*. AAAI Press, (1998).

[20] G. Towell and E. M. Voorhees, 'Disambiguating Highly Ambiguous Words', *Computational Linguistics*, **24**(1), 125–146, (1998).

[21] D. Yarowsky, 'Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French', in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*. ACL, (1994).