Machine Learning and Natural Language Processing

Lluís Màrquez

Centre de recerca TALP
Departament de Llenguatges i Sistemes Informàtics, LSI
Universitat Politècnica de Catalunya, UPC
Jordi Girona Salgado, 1–3
E-08034, Barcelona
lluism@lsi.upc.es

July 11, 2000

Abstract

In this report, some collaborative work between the fields of Machine Learning (ML) and Natural Language Processing (NLP) is presented. The document is structured in two parts. The first part includes a superficial but comprehensive survey covering the state–of–the–art of machine learning techniques applied to natural language learning tasks. In the second part, a particular problem, namely Word Sense Disambiguation (WSD), is studied in more detail. In doing so, four algorithms for supervised learning, which belong to different families, are compared in a benchmark corpus for the WSD task. Both qualitative and quantitative conclusions are drawn.

This document stands for the complementary documentation for the conference "Aprendizaje automático aplicado al procesamiento del lenguaje natural", given by the author within the course: "Curso de Industrias de la Lengua: La Ingeniería Lingüística en la Sociedad de la Información", Fundación Duques de Soria. Soria. July 2000.

Contents

1 Introduction

In the eighties but especially throughout the nineties, an important resurgence of empirical and statistical methods, applied to the automatic processing of natural language, has been observed (under the name of *corpus–based*, *statistical*, or *empirical* methods). In the last five to ten years there has been a dramatic shift in computational linguistics from manually constructing grammars and knowledge bases to partially or totally automating this process by using statistical learning methods trained on large annotated or unnanotated natural language corpora. This popularity has its origin in four main factors:

- The growing availability of big machine–readable corpora, from different sources, levels of annotation, languages, etc.
- The improvement in performance of current software and hardware architectures that allow the processing of huge amounts of information, by highly time–consuming algorithms (as statistical models usually demand), at an admissible cost of time and money.
- The initial success obtained with the statistical processing of low–level language problems, such as those related with speech recognition and syntactic tagging.
- The appearance and development of a large amount of text–based natural language applications with specific requirements, for which conventional methods based on linguistic knowledge seemed not to be appropriate.

The choice of automatically processing such massive quantities of free text (commonly referred to *data–intensive approach* or *corpus–based approach*) has contributed to developing a number of methods and techniques with an application to a great variety of natural language acquisition and understanding problems, including: automatic extraction of lexical knowledge, lexical and structural disambiguation (part–of–speech tagging, word sense disambiguation and prepositional phrase attachment disambiguation), grammatical inference and robust parsing, information extraction and retrieval, automatic summarization, machine translation, etc. Additionally, corpora have provided linguists with benchmarks for the empirical evaluation of theoretical studies and models of language. The reader may find good and plain introductions to the corpus–based NLP in [114, 46, 244].

Most of the initial corpus–based language acquisition methods applied by the NLP community researchers were borrowed from statistics and information theory. As a consequence of this collaboration, a significant progress was made in the development and adaptation of well–known statistics based techniques to the particular problems of automatic natural language modelling and processing. This progress was especially noticeable in the low–level tasks of speech processing and lexical knowledge extraction and disambiguation, but a considerable effort was also devoted to grammatical inference, robust parsing and other semantic and discourse level NLP tasks.

Several articles and introductory books, surveying statistical approaches in Computational Linguistics, have appeared in recent years. We especially recommend [41, 42, 106, 28, 102].

Starting in the early 90s, but particularly in recent years, the application of machine learning (ML) based techniques to language learning and acquisition problems has been the focus of increasing attention in the NLP community. The core of problems addressed by machine learning techniques are those of natural language **disambiguation** (appearing at all levels of the language understanding process). They are particularly appropriate because they can be easily recast as *classification problems*, a generic type of problem with a long tradition in the artificial intelligence (AI) area, and especially addressed by the ML community.

Already applied ML based methods include several *traditional* symbolic inductive learning paradigms: instance–based learning, decision trees, threshold linear separators, inductive logic, unsupervised clustering, etc.; and also a number of subsymbolic and connectionist approaches, such as neural networks and genetic algorithms. Additionally, some new specific learning algorithms have also been proposed in recent years. This is the case of *Transformation–based Learning* in [19]. This recent approach permits the use and adaptation of general purpose machine learning algorithms for classification –which are well known and widely studied methods–, with the aim of providing general frameworks in which many disambiguation problems could be addressed simultaneously by homogeneous techniques [19, 31, 59, 188].

Some concrete publication statistics clearly illustrate the extent of the revolution in natural language research. As an example, a full 63.5% of the papers in the Proceedings of the Annual Meeting of the Association for Computational Linguistics and 47.7% of the papers in the journal *Computational Linguistics* concerned corpus–based research in 1997 (the percentages of 1990 were still only 12.8% and 15.4%). Furthermore, the connection and collaboration between NLP and machine learning communities is becoming noticeable in the international conferences and journals of both areas, which have experimented a proliferation of special issues on *natural language learning*, *applications of ML techniques to natural language processing*, etc.

Despite this growing collaboration, most of the research performed on natural language learning has been carried out within the Computational Linguistics research community. This is unfortunate, since we think that machine learning and empirical NLP have much to offer each other and that increased interaction and exchange of ideas would greatly benefit both areas.

On the one hand, a variety of thoroughly studied and well–understood alternative algorithms can be applied to natural language problems and can have significant advantages in particular applications. In addition to specific learning algorithms, a variety of general ideas from traditional machine learning such as "active learning", "boosting", "reinforcement learning", "constructive induction", "learning with background knowledge", "theory refinement", "experimental evaluation methods", etc. may also be usefully applied to natural language problems.

On the other hand, natural language can provide machine learning with a variety of interesting and challenging problems, frequently with particular characteristics such as: Very large feature space, extremely sparse data, presence of irrelevant and highly correlated features (redundancy), noisy training data, very large training sets, necessity of unsupervised or semi–supervised learning (when training data is difficult to obtain), problems that go beyond the simple classification scheme, e.g. parsing, in which a structured parse tree should be generated, etc. Further, as already said, some learning paradigms have been introduced by empirical NLP, e.g. TBL [21, 19] (Transformation-based Error-Driven Learning), Maximum Entropy [110, 187, 179], etc. These methods are not well studied in the machine learning literature. Perhaps these methods could also be productively applied to other machine learning problems.

For an extensive compilation of relevant articles about the current approaches to natural language learning, one may consult [231, 32].

The Ambiguity Problem

Natural language is ambiguous in nature. Ambiguity appears at many levels of the usual language processing chain, and it represents many of the most difficult problems of language understanding. Some classical examples of ambiguity are: Word selection in speech recognition, part of speech ambiguity (e.g. past vs. participle in regular verbs), semantic ambiguity in polysemic words, structural ambiguity in parsing (e.g. PP–attachment), accent restoration, reference ambiguity in anaphora resolution, word choice selection in machine translation, context–sensitive spelling correction, etc.

Such *ambiguity resolution* problems can be generically characterized as follows: *At a certain moment, the* NLP *system reaches a segment of information for which it has multiple interpretations, and it must decide which interpretation is appropriate in the current context. In order to resolve this difficulty, it is necessary to disambiguate two or more semantically, syntactically or structurally distinct forms based on the properties of the surrounding context* [31].

Consider, for instance, the following simple sentence, which was picked from the Wall Street Journal (WSJ) corpus:

He was shot in the hand as he chased the robbers in the back street. (1)

First, the sentence contains a number of PoS ambiguities that should be resolved before the sentence can be understood. For instance, "shot" and "hand" can be a noun or a verb; "chased" can be an adjective or a verb; "back" can be a noun, an adjective, an adverb, and a verb; finally, "in" and "as" can be a preposition or an adverb. Even if we know its part of speech, the intended meaning of the word in a particular context often requires disambiguation. In the present example, the word "hand" is highly polysemous. Among other interpretations, it could refer to a part of the body in the *anatomical* sense, but also to a part of a clock in a *mechanical* sense. In addition to these cases of lexical ambiguity we also find an example of structural ambiguity: The prepositional phrase "in the back street" could modify the noun "robbers" or the verb "chased". Both readings are syntactically legal and the NLP system must access some kind of semantic information to make the correct attachment decision.

Overview of this document

Section 2 surveys state–of–the–art main approaches related with the application of machine learning techniques to natural language learning and disambiguation tasks. Section 3 is devoted to explain a thorough comparative experiment between four machine learning algorithms for supervised word sense disambiguation. Note that, after the references and acknowledgements, some additional material has been included. Appendix A contains the technical details about the algorithms tested, which have not been included in section 3 (mainly, for clarity reasons).

2 A survey on Machine Learning Techniques Applied to Natural Language Processing

In this section, a broad-coverage compilation of references linking the fields of machine learning and natural language processing is provided. Although in a very compact way, we think that it contains valuable information that can contribute to ease the approach of any researcher to the use of machine learning techniques in processing natural language[1].

Learning approaches are usually categorized as **statistical** (also probabilistic or stochastic) methods and **symbolic** methods, belonging to the latter the typical learning paradigms that do not explicitly use probabilities in the hypothesis (decision trees, instance–based learning, rule–induction systems, etc.)[2]. We will follow this criterion in the following exposition merely for clarity reasons. Additionally, we have treated separately the subsymbolic and connectionist approach, and we have included a last category, containing unsupervised approaches (all other referenced methods belong to the supervised family) and the recently emergent approach of combining classifiers. The main focus will be on the symbolic family.

2.1 Stochastic Machine Learning Approaches

Dietterich [65] define a *stochastic model* as a model that describes the real–world process by which the observed data are generated. The stochastic models are typically represented as a probabilistic network that represents the probabilistic dependencies between random variables. Each node in the graph has a distribution, and from these individual distributions, the joint distribution of the observed data can be computed. Different approaches vary in how this probabilistic network is acquired and in which is the method applied to combine individual probability distributions.

The most simple approach to stochastic classification is to use the Naive Bayes Classifier (NB), originally described in [68], which is based on the Bayes' theorem and the assumption of independence between features. Despite its simplicity it has been widely used in the ML and NLP communities with a surprising success. NB provides a simple way to combine information from several sources, however, when the statistical sources to combine are of different degree of generalization NB is usually combined with back–off estimates. We can find the NB algorithm (either the basic version or other variations and hybrids) applied to the following NLP disambiguation tasks: Context–sensitive spelling correction [88, 89], PoS tagging [194, 189], PP–attachment disambiguation [52], Word sense disambiguation [86, 150, 156, 112, 73] and Text Categorization [117, 190, 119, 142, 196].

Recently, Lau, Rosenfeld and Roukos [110, 187] have proposed a new approach for combining statistical evidence from different sources, that is based on the *Maximum Entropy Principle* (ME). This work was originated within the speech recognition field [187], but it has also been successfully applied to word morphology [168], PoS tagging [101, 177], PP–attachment disambiguation [180], identification of clause boundaries [181], partial and general parsing [211, 178], text categorization [161], and machine translation [10]. See [179] for a broad introduction to ME methods and a survey of existing applications.

Hidden Markov Models [175], already referenced in the previous section, can be also seen as stochastic models of learning. HMMs had their major success in the low-level tasks of language disambiguation, that is, speech recognition and synthesis [175, 97], PoS tagging [45, 54, 144], and named entity recognition and classification [11]. However, there have been also efforts to

[1] Those interested can consult the following URL to find a larger list of related references: `http://www.lsi.upc.es/~lluism/BibTexDB.html`. The available format is BibTex and several links are provided to help obtain postscript versions of the original papers.

[2] However, as Roth points in [188], all learning methods are statistical in the sense that they attempt to make inductive generalization from observed data and use it to make inferences with respect to previously unseen data.

extend the use of HMM to WSD [205], and partial parsing by tagging grammatical functions and bracketing simple constituents [16, 212].

The *Expectation Maximization* (EM) algorithm is an iterative algorithm that starts with an initial value for the parameters of the model and incrementally modifies them to increase the likelihood of the observed data [63]. Particular instances of the EM algorithm have been applied to a number of different problems in NLP. For instance, the parameter estimation of HMM models is done by the EM algorithm, namely the Baum–Welch or *Forward–Backward* algorithm [41]. The *Inside–Outside* algorithm [41] is another version of the EM algorithm related to the estimation of parameters for probabilistic grammars (Stochastic Context–Free Grammars, Stochastic Lexicalized Tree–Adjoining Grammars, etc.) and to the grammar inference from annotated corpus to produce robust parsers [167, 25, 41, 120]. The EM algorithm is also used in the *Linear Interpolation* approach for smoothing in HMM–based models [41, 132], in a version of unsupervised sense discrimination by Schütze [203], in semi–supervised word sense disambiguation [166], and, in combination with the Naive Bayes classifier, in a semi–supervised approach to text classification [162, 163].

Finally, *log–linear* models [44] are also being applied to natural language processing. In particular, log–linear regression [195], a popular technique for binary classification, is used in [209] to classify verbs for machine translation purposes. In the same direction, the work by Marques et al. [135] use log–linear models to induce verbal transitivity.

2.2 Symbolic Machine Learning Approaches

2.2.1 Decision Trees

Decision tree based methods of supervised learning from examples represent one of the most popular approaches within the AI field for dealing with classification problems [17, 170, 171, 173].

Decision trees are a way to represent rules underlying training data, with hierarchical sequential structures that recursively partition the data. They have been used for years in several disciplines such as statistics, engineering (pattern recognition), decision theory (decision table programming), and signal processing. More recently renewed interest has been generated by the research in artificial intelligence (machine learning, expert systems, etc.). In all these fields of application, decision trees have been used for data exploration with some of the following purposes[3]: *Description* (i.e. to reduce a volume of data by transforming it into a more compact form), *Classification* (i.e. discover whether the data contains well–separated and meaningful clusters of objects) and *Generalization* (i.e. uncovering a mapping from independent to dependent variables that is useful for predicting the value of the dependent variable in the future).

Their application to NLP is also noticeable, and we find tree–based solutions to address natural language ambiguity problems at several levels: Speech recognition [8, 9], PoS tagging [200, 132, 140, 141, 164, 136, 138, 137], word sense disambiguation [26], parsing [132, 92], text categorization [117, 69, 230], text summarization [134], dialogue act tagging [192], co–reference resolution [5, 143], cue phrase identification [122], and machine translation (verb classification) [221, 209].

In Magerman's approach [132], decision trees are used for a number of simultaneous different decision–making problems, such as: Assigning part–of–speech tags to words, assigning

[3] This classification is extracted from [154].

constituent labels, determining the constituent boundaries in a sentence, deciding the scope of conjunctions, etc. In a previous work [12] a mixed, statistical and tree–based, approach was used to pick up the correct parse among all possibilities. Other mixed approaches are that of Schmid [200] and Kempe [103] who introduced decision trees for estimating the transition probabilities in HMM–based taggers.

In concept–learning, decision trees are sometimes translated into rules (and eventually pruned) for representing the target concept. The most representative system is C4.5-RULES, a variant of C4.5 [173], which is used for instance in [134] for automatic summarization.

There are other popular logic–based rule–induction systems that employ different representations of concepts: Disjunctive normal form (DNF), conjunctive normal form (CNF) and decision lists. The FOIL algorithm [172] and some variants [149, 234] have been widely used to acquire first–order logic representations, and, in relation to the NLP classification problems, we find them tested in many of the already cited works as benchmarks for any other applied inductive learning algorithm.

2.2.2 Decision Lists

Decision lists [184] are ordered lists of conjunctive rules (where rules are tested in order and the first one that matches an instance is used to classify it) which have been applied in a number of concept–learning systems [47, 152, 174].

Decision lists work well in domains with many attributes (or with attributes with many values) because they avoid to some extent the data fragmentation problem. Thus, regarding NLP, they have been applied to lexical ambiguity resolution. In particular: Word sense disambiguation, lexical choice in machine translation, homograph disambiguation in speech synthesis and accent restoration [240, 243, 241, 150, 208].

2.2.3 Transformation–Based Error–Driven Learning (TBL)

TBL was introduced by Brill in the early 90s, as a new approach to corpus–based natural language learning. The learning algorithm is a mistake–driven greedy procedure that produces a set of rules. It works iteratively by adding at each step the rule that best repairs the current errors. Concrete rules are acquired by instantiation of a predefined set of template rules.

This algorithm has been applied to a number of natural language problems, including part–of–speech tagging [21, 22, 23, 6, 186], PP–attachment disambiguation [20], parsing [18], spelling correction [133], and word sense disambiguation [67].

One major drawback of TBL is its computational cost since all instantiations of templates are tested at each iteration to find the best rule. Recently, Samuel [191] presented an efficient approximation called Lazy TBL which restrict the search to a small subset of all possible instantiations, by applying Monte Carlo sampling techniques, with a very slight decrease in accuracy. In this way, more complex problems can be faced using LTBL. In particular, Samuel and colleagues have applied this new algorithm to dialogue act tagging [192, 193], that is, to label each utterance in a conversational dialogue with the correct *dialogue act*, which is a concise abstraction of a speaker's intention.

Another rule–learning system successfully applied to text categorization is Cohen's RIPPER algorithm [51]. In this case, the algorithm learns a classifier in the form of a boolean combination of simple terms.

2.2.4 Linear Separators

Linear separators with multiplicative weight–update algorithms[4], have been shown to have exceptionally good behaviour when applied to very high dimensional problems, in the presence of noise, and specially when the target concepts depend on only a small subset of the features in the feature space. Clearly, this is a usual scenario in the text processing domain. Roth and colleagues have designed the SNOW architecture [188], a sparse network of linear separators in the feature space, using the WINNOW algorithm [123], for on–line and adaptive learning. They have applied it successfully to a broad spectrum of natural language disambiguation tasks, including context–sensitive spelling correction [89], PoS tagging [189], PP–attachment disambiguation [107], shallow parsing [130], text categorization [62], and word sense disambiguation [74], achieving state–of–the-art accuracies and surpassing several alternative algorithms.

Other methods based on linear separators have been applied to the text categorization task. Cohen and Singer [51] presented EXPERTS (based on the weighted majority algorithm [124]), Lewis et al. [118, 64] used *Widrow–Hoff* [232] and EG (Exponentially Gradient) [105] algorithms to text categorization and routing. Again, another variation on the WINNOW algorithm, BALANCEDWINNOW$^+$ [62], has reported competitive results in the text categorization task. All these algorithms proved to overcome one of the most commonly used techniques, the Rocchio algorithm and variants [185, 91, 118, 196, 64], which is a classifier that uses vectors of numeric weights to represent the data (vector space model) and works in a relevance feedback context.

2.2.5 Instance–based Learning

Instance–based learning algorithms [3, 4, 126] have appeared in several areas of the AI with many different names: Similarity–based, example–based, case–based, memory–based, exemplar–based, analogical, etc. It is a form of supervised, inductive learning from examples, based on keeping full memory of training material and classifying new examples by using a sort of k–nn (k nearest neighbours) algorithm.

We find several uses of this kind of algorithms in NLP tasks. Particularly relevant is the work by Cardie [36, 38], which addressed the lexical, semantic and structural disambiguation of full sentences (in limited domains), within an information extraction (IE) environment. Additionally, her instance–based system takes advantage of decision–trees for identifying relevant attributes [37]. More recent work refers to relative pronoun resolution [39], and to the description of the *Kenmore Framework* [31], a general framework that embedded machine learning algorithms for a global treatment of many natural language problems.

Equally essential is the recent work of the ILK Group at Tilburg University. Daelemans and colleagues have developed the TiMBL (Tilburg Memory–based Learning Environment) which is a general instance–based algorithm that makes a compression of the base of examples into a tree–based structure, IGTree [57], used later for classifying new examples. These trees have proved to reduce significantly the space requirements and to be very efficient and accurate in several domains [58], including: Phonology (stress, word pronunciation) and morphology

[4]Linear threshold algorithms, like WINNOW, are very simple on–line learning algorithms for 2–class problems with binary (i.e., 0/1–valued) input features. To classify new examples, they simply calculate a weighted sum of input features (linear combination) and outputs 0 if the result is below the threshold, and 1 otherwise. Wrongly predicted training examples make the weights of the model change, in a multiplicative way, to better fit the training set.

[55, 14, 56, 15], PoS tagging [61, 60, 90], PP–attachment disambiguation [246], shallow parsing [227], and smoothing of probability estimates [245].

The work of other authors include applications to partial parsing (chunking) and context–sensitive parsing [210, 7, 33], WSD [159, 157, 84, 73], text categorization [183, 238, 237, 239], semantic interpretation [38], machine translation [100], and lexical acquisition by analogy [76, 75].

2.2.6 Inductive Logic Programming (ILP)

This is a discipline devoted to the inductive learning of Prolog programs from examples. The most relevant work in relation to natural language learning has been carried out by Mooney and colleagues at the University of Texas. A general survey of applications of ILP to NLP can be found in [151]. Particular works include applications to grammatical inference [247, 248], automatic induction of natural language interfaces for querying data bases [249, 222], information extraction tasks [216, 217, 29, 79, 80, 81, 30, 215], acquisition of verbal properties [153], text categorization [49, 50, 53, 213], and generation of natural language [176].

2.3 Subsymbolic Machine Learning Approaches

2.3.1 Neural Networks

In their relation to NLP, neural networks [94] have been used basically to address low–level problems, such as OCR [204], speech recognition and synthesis [206, 121, 155, 113, 229], and PoS tagging [155, 201, 70, 199, 131]. The basic models refer to *feed–forward* multilayer neural networks trained with the *backpropagation* algorithm, but also include some examples of recurrent networks and ensembles of several single neural networks.

Other examples addressing more complex problems, sometimes in combination with symbolic approaches, are: Identification of clause boundaries [95], parsing and sentence analysis [115, 43, 128, 129], grammatical inference [111], PP–attachment disambiguation [218, 125], WSD [224], text categorization [233], and detecting spelling errors [116]. In [125] there is a present day survey of neural networks with application to NLP[5].

2.3.2 Genetic Algorithms

Genetic Algorithms [87, 219] have been basically used in language learning problems [214] from a non informed perspective, that is trying to infer word categories and syntactic structure from the sole source of unannotated corpora and with no a priori knowledge. This is an approach that it has been also addressed with unsupervised learning algorithms for clustering. Relevant contributions here are: [108, 109, 214] and [127], in which language learning is approached within an information retrieval and filtering framework. The work by Yang [236] also applies genetic algorithms to information retrieval. Finally, in [209] we find an application to verbal classification for machine translation purposes.

[5] This survey is written in Spanish.

2.4 Others

2.4.1 Clustering Algorithms

Concept formation and clustering algorithms are instances of the unsupervised machine learning paradigm [78, 77]. They have been used in the NLP field in tasks such as: Document retrieval, automatic hyphenation, semantic, syntactic and phonological classification, extraction of hierarchical structure, machine translation, etc. See [169] for a good survey including pointers to relevant references. Additionally, conceptual clustering algorithms have been used by Cardie [35, 34] to tackle relative pronoun and noun phrase coreference resolution in a information extraction framework.

2.4.2 Ensembles of Classifiers

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (usually by weighted or unweighted voting) to classify new examples. These techniques have been mainly studied in the supervised learning area and it has been proven that ensembles are often much more accurate than the individual classifiers that make them up. Within the voting (and variants) approaches, and related to NLP problems, we find ensembles applied to part–of–speech tagging [90, 24, 139, 136], context–sensitive spelling correction [89], word sense disambiguation [182, 165], shallow parsing [223], and anaphora resolution [148, 147][6].

More complex combination strategies, and algorithms for constructing the ensembles, including stacking, bagging, boosting, etc. can be found in text categorization [196] and text filtering [198], where an adapted version of the popular AdaBoost algorithm [82, 83, 197] is presented for information retrieval tasks. Other applications of AdaBoost variants to NLP tasks include: PoS tagging [2], PP–attachment disambiguation [2], word sense disambiguation [72, 74], and full parsing [93]. [13] is another relevant example of the combination of classifiers applied to information retrieval, in which the combination of classifiers allows the use of a big set of unlabelled examples (semi–supervised approach) to iteratively improve the classification accuracy in the task of filtering web pages.

2.4.3 Support Vector Machines

Support Vector Machines (SVM) were introduced by Vapnik in 1979 [225], but they have only recently been gaining popularity in the learning community. SVMs are based on the *Structural Risk Minimization* principle from Computational Learning Theory [225, 226] and in their basic form, they learn the linear hyperplane that separates a set of positive examples from a set of negative examples with maximum margin (the margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples). Despite the lineality of the basic algorithm, by using appropriate kernel functions, SVMs can be extended to learn polynomial classifiers, radial basic function (RBF) networks, and three–layer sigmoid neural nets.

SVMs have been succesfully applied to a number of problems related to pattern recognition. Regarding NLP, SVMs have obtained the best results to date in Text Categorization [98, 69]. Additionally, a slightly different approach, called Transductive Support Vector Machines allows a semi–supervised learning procedure, which is very useful in domains with few training

[6] In these cases, the disambiguation is performed by straightforwardly combining a set of pre–existing classifiers, heuristics, or predictors.

instances but a very large test set (Relevance Feedback, Netnews Filtering, Reorganizing a document collection, etc.). In [99] it is empirically shown that transductive learning has advantages over the standard inductive approach in the text classification domain.

2.5 A Reversed Summary

The already described survey about the interactions between the fields of NLP and ML has been organized by taking as a reference the machine learning paradigms and showing which NLP problems have been addressed by each of them. In the current section, we present a reversed summary, that is, indexing the information by the type of NLP task to be solved. The information is presented in the form of tables.

The notation used for the machine–learning algorithms appearing in the tables is the following: DTs stands for Decision Trees, HMMs stands for Hidden Markov Models and statistical approaches, ME stands for the Maximum Entropy Approach, IBL stands for Instance–Based (or memory–based) Learning, NNs stands for Neural Networks, TBL stands for the Transformation–Based (error–driven) Learning, NB stands for Naive Bayes classifiers and derived approaches, LSM stands for Linear Separators (on–line classifiers with Multiplicative updating functions), GAs stands for Genetic Algorithms, Clust stands for Clustering algorithms, DLs stands for Decision Lists, ILP stands for Inductive Logic Programming, Rocchio stands for Rocchio's algorithm for text categorization, RI stands for Rule Induction algorithms, EC stands for any algorithm that uses ensembles of classifiers or simple combination of heuristics, SVM stands for Support Vector Machines, and, finally, LogL stands for Log–linear Models.

Table 1 contains information about low–level NLP tasks, such as speech processing, morphology and PoS tagging.

	NB	DTs	HMMs	ME	TBL	NNs
Speech recognition and synthesis		[8, 9]	[97]	[187]	[55, 56, 15]	[206, 121, 155, 113, 229]
Morphology					[14]	
PoS tagging	[194, 189]	[200, 132, 140, 141, 164, 136, 138, 137]	[45, 54, 144]	[101, 177]	[21, 22, 23, 6, 186]	[155, 201, 70, 199, 131]

	IBL	LSM	EC			
PoS tagging	[61, 60, 90, 58]	[188]	[90, 24, 139, 2, 136]			

Table 1: References corresponding to some low–level NLP tasks

Table 2 contains the references about parsing (either shallow or general) and structural ambiguity resolution.

Table 3 groups the references about semantic and discourse–level NLP tasks, namely, sense disambiguation, co–reference resolution, anaphora resolution, dialogue act tagging, and text filtering and categorization, which are NLP tasks usually associated to information retrieval and information extraction.

Table 4 summarizes the references corresponding to different levels (lexical, syntactic, semantic, etc.) of language acquisition.

	DTs	HMMs	ME	IBL
Clause Boudaries			[181]	
Shallow Parsing		[45, 1, 16, 212]	[211]	[7, 227, 33, 58]
Parsing	[12, 132, 92]		[178]	[210, 37, 36, 38]
PP–attachment disambiguation			[180]	[246]

	TBL	NB	NNs	LSM	EC
Clause Boudaries			[95]		
Shallow Parsing	[18]		[128, 129]	[130]	[223]
Parsing			[115, 43]		[93]
PP–attachment disambiguation	[20]	[52]	[125, 218]	[107]	[2]

Table 2: References corresponding to syntactic analysis and structural ambiguity NLP problems

	DLs	DTs	NB	TBL	EM
WSD	[240, 150]	[26, 150]	[86, 150, 112]	[67]	[203, 166]
Text categorization and filtering		[117, 69, 230]	[117, 190, 119, 142, 196]		[162, 163]
Dialogue act tagging		[192]		[193, 192]	
Co–reference and anaphora resolution		[5, 143]			
Cue phrase identification		[122]			

	IBL	NNs	EC	SVM	Clust
WSD	[159, 157, 84, 73]	[150, 224]	[182, 72, 74, 165]		[202]
Text categorization and filtering	[183, 238, 237, 239]	[233]	[198, 196, 13]	[98, 69, 99]	
Co–reference and anaphora resol.	[39]		[148, 147]		[35, 34]

	Rocchio	RI/ILP	LSM	GAs	ME
WSD			[74]		
Text categorization and filtering	[185, 91, 118, 196, 64]	[49, 51, 50, 53, 134, 213]	[51, 118, 62, 64]	[236, 127]	[161]
Information Extraction		[216, 217, 29, 79, 80, 81, 30, 215]			

Table 3: References corresponding to the discourse–level semantics NLP problems.

	IBL	ILP	NNs	GAs	Clust
Lexical acquisition	[76, 75]				[169]
PoS acquisition				[108, 127]	
Grammatical Inference		[247, 248, 249, 222]	[111]	[109, 127, 214]	
Semantic acquisition	[38]				[169]

Table 4: References corresponding to automatic language inference tasks.

Finally, table 5 contains references about other NLP tasks, such those related with machine translation, spelling correction, etc.

	DTs	ME	IBL	TBL	NB
Acquisition of verbal properties	[221, 209]				
General machine translation		[10]	[100]		
Spelling correction				[133]	[86, 88, 89]

	DLs/ILP	NNs/Clust	GAs	LSM	LogL
Acquisition of verbal properties	[152, 153, 29]		[209]		[209, 135]
General machine translation		[235]			
Spelling correction	[241, 208]	[116]		[89]	
Generation	[176]				

Table 5: References corresponding to Machine Translation and other NLP tasks

3 Word Sense Disambiguation: A Case Study in Supervised Machine Learning

The present section is devoted to explain the comparison between four machine learning algorithms applied to Word Sense Disambiguation. This work, which can be also found in [74], is organized as follows: Sections 3.1 and 3.2 describes the word sense disambiguation problem. Section 3.3 defines the general framework of supervised learning for classification and section 3.4 presents the four ML algorithms used. In section 3.5 the general setting is presented, including the corpora and the experimental methodology used. Sections 3.6, 3.7, and 3.8 report the experiments carried out and the results obtained. Finally, section 3.9 concludes and discusses some lines for further research.

3.1 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the problem of assigning the appropriate meaning (sense) to a given word in a text or discourse where this meaning is distinguishable from other senses potentially attributable to that word. As an example, table 6 shows the definition of two senses of the word *age*[7] and an example sentence[8] for each sense. Thus, a WSD system must be able to assign the correct sense of a given word, for instance *age*, depending on the context in which the word occurs.

Sense Definitions	
age 1	the length of time something (or someone) has existed; "his age was 71"; "it was replaced because o its age"
age 2	a historic period; "the Victorian age"; "we live in a litigious age"

Corpora Examples	
age 1	He was mad about stars at the **age** of nine .
age 2	About 20,000 years ago the last ice **age** ended .

Table 6: Sense definitions and corpora examples.

Resolving the ambiguity of words is a central problem for Natural Language Processing (NLP) applications and their associated tasks [96], including, for instance, natural language understanding, machine translation, information retrieval and hypertext navigation, parsing, acquisition of subcategorization patterns, selectional restrictions, speech synthesis, spelling correction, reference resolution, automatic text summarization, etc.

WSD is one of the most important open problems in NLP. Despite the wide range of approaches investigated [104] and the large effort devoted to tackle this problem, it is a fact that to date, no large-scale broad-coverage and highly accurate WSD 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 previously semantically annotated corpus) have obtained better results than unsupervised methods on small sets of selected ambiguous words, or artificial pseudo-words.

[7] The senses of the example have been taken from the WordNet 1.5 [146].
[8] These examples have been taken from the DSO corpus [159].

Many standard ML algorithms for supervised learning have been applied to WSD, including: Decision Lists [242], Neural Networks [224], Bayesian learning [27], Exemplar Based learning [156, 84], and Boosting [72]. Further, in [150] some of the previously cited methods are compared, jointly with Decision Trees and Rule Induction algorithms, on a very restricted domain.

The performance of supervised ML based systems is usually calculated by testing the algorithm on a separate part of the set of annotated examples (say 10–20%), or by N–fold cross–validation, in which the set of examples is partitioned into N disjoint sets (or folds), and the training–test procedure is repeated N times using all combinations of $N-1$ folds for training and 1 fold for testing. In both cases, test examples are different from those used for training, but they belong to the same corpus, and, therefore, they are expected to be quite similar.

Although this methodology could be valid for certain NLP problems, such as English Part-of-Speech tagging, we think that there exists reasonable evidence to say that, in WSD, accuracy results cannot be simply extrapolated to other domains:

- WSD is very dependant on the domain of application. In [85], the idea of "one sense per discourse" has been suggested, that is, if a polysemous word appears two or more times in a well-written discourse, it is extremely likely that they will all share the same sense.
- Further, see [159, 156], where quite different accuracy figures are obtained when testing an Exemplar Based WSD classifier on two different corpora, namely Wall Street Journal and Brown Corpus (both corpora belong to different domains).
- It does not seem reasonable to think that the training material is large and representative enough to cover "all" potential types of examples.

We think that the study of the domain dependence of WSD —in the style of other studies devoted to parsing [207]— is needed to assess the validity of the supervised approach, and to determine to which extent a tuning process is necessary to make real WSD systems portable. In order to corroborate the previous hypotheses, this work explores the portability and tuning of four different ML algorithms by training and testing them on different corpora.

3.2 Is Supervised WSD Really Viable?

It is well-known that supervised methods suffer from the lack of widely available semantically tagged corpora, from which to construct really broad coverage WSD systems. This is known as the "knowledge acquisition bottleneck" [86]. In [158] is estimated that the manual annotation effort necessary to build a broad coverage semantically annotated English corpus would be about 16 man–years. This extremely high overhead for supervision (which could be much greater if a costly tuning procedure is required before applying any existing system to each new domain) 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, the need for supervision, and the computational requirements in corpus-based methods for WSD. Consequently, the following four lines of research are being explored:

1. The design of efficient sampling methods [71, 84];
2. The use of external lexical resources, such as WordNet [146], and Web search engines to automatically obtain from Internet arbitrarily large samples of word senses [112, 145];
3. The use of unsupervised EM-like algorithms for estimating the statistical model parameters [166].
4. The application of efficient and accurate ML algorithms [72] and attribute representations [73] in order to deal with real size WSD problems.

Solving the problem of knowledge acquisition is crucial for making the unsupervised WSD approach viable. It is our belief that the referred work, and especially second and fourth lines, provides enough evidence towards the "opening" of the bottleneck in the near future. For that reason, it is worth further investigating the robustness and portability of existing supervised ML methods to better resolve the WSD problem.

3.3 Machine Learning Framework: Supervised Learning for Classification

The goal in supervised automated learning for classification consists of inducing an approximation (or *hypothesis*) of an unknown function f defined from an input space Ω to a discrete unordered output space[9]: $\{1, \ldots, K\}$, given a set of training examples:

$$T = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))\}.$$

The components of each example $\mathbf{x}_i$ are typically vectors of the following form:

$$\mathbf{x}_i = \langle x_{i,1}, x_{i,2}, \ldots, x_{i,m} \rangle,$$

whose components, called *features* (or *attributes*) of $\mathbf{x}_i$, are discrete or real–valued. Therefore the objects of the domain are completely described by a set of attribute–value pairs, and a class label.

The function $f : \Omega \rightarrow \{1, \ldots, K\}$ defines a K-partition of the input space into sets $f^{-1}(k)$ called *classes* and denoted y_k.[10]

Given a training set T, a learning algorithm outputs a classifier, denoted h, which is a hypothesis about the true function f. This process of deriving classification rules from samples of classified objects is sometimes called *discrimination*. Given new $\mathbf{x}$ values, h predicts the corresponding y values, i.e. it *classifies* the new examples.

3.4 Learning Algorithms Tested

This section provides some references and short descriptions of the main characteristics of the four learning methods compared in this report: Naive Bayes, Exemplar Based, SNoW and LazyBoosting.

[9] When a continuous output space is considered we talk about *regression* instead of *classification*.

[10] Sometimes f is viewed as a distribution (and not a deterministic mapping). In this more general situation f^{-1} would be ill–defined. Nevertheless, we state it as deterministic for the sake of simplicity.

3.4.1 Naive Bayes (NB)

Naive Bayes is intended as a simple representative of statistical learning methods. It has been used in its most classical setting [68]. That is, assuming independence of features, it classifies a new example by assigning the class that maximizes the conditional probability of the class given the observed sequence of features of that example.

Let $\{1 \dots K\}$ be the set of classes and $\{x_{i,1}, \dots, x_{i,m}\}$ the set of feature values of a example $\mathbf{x}_i$. The Naive Bayes method tries to find the class that maximizes $P(k \mid x_{i,1}, \dots, x_{i,m})$:

$$\arg\max_k \; P(k \mid x_{i,1}, \dots, x_{i,m}) \approx \arg\max_k \; P(k) \prod_j P(x_{i,j} \mid k) \, ,$$

where $P(k)$ and $P(x_{i,j} \mid k)$ 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 [156], has been used. It consists in replacing zero counts of $P(x_{i,j} \mid k)$ with $P(k)/N$ where N is the number of training examples.

Despite its simplicity, Naive Bayes is claimed to obtain state–of–the–art accuracy on supervised WSD in many papers [150, 156, 112].

3.4.2 Exemplar Based Classifier (EB)

In Exemplar Based learning [220, 3] no generalization of training examples is performed. Instead, the examples are stored in memory and the classification of new examples is based on the classes of the most similar stored examples. After the instance base is built, new (test) instances are classified by matching them to all instances in the instance base, and by calculating with each match the *distance* between the new instance $\mathbf{x} = \langle x_1, \dots, x_m \rangle$ and the stored instance $\mathbf{y} = \langle y_1, \dots, y_m \rangle$.

The most basic metric for instances with symbolic features is the overlap metric (also called Hamming distance), defined as follows:

$$\Delta(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} w_i \delta(x_i, y_i),$$

where w_i is the weight for i_th feature and $\delta(x_i, y_i)$ is the distance between two values, which is defined as:

$$\delta(x_i, y_i) = 0 \;\text{ if }\; x_i = y_i, \;\; 1 \;\text{ otherwise}\, .$$

In our basic implementation, all examples are stored in memory and the classification of a new example is based on a k–NN (Nearest–Neighbours) algorithm using Hamming distance to measure closeness (in doing so, all examples are examined and no feature weighting is used, i.e. $w_i = 1$ for all i). For k's greater than 1, the resulting sense is the weighted majority sense of the k nearest neighbours —where each example votes its sense with a strength proportional to its closeness to the test example.

In the experiments explained in following sections, the EB algorithm is 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[11].

[11] In order to construct a real EB–based system for WSD, the k parameter would be estimated by cross–validation using only the training set [156].

Exemplar Based learning is said to be the best option for WSD [156]. Other authors [58] point out that Exemplar Based methods tend to be superior in language learning problems because they do not forget exceptions.

3.4.3 SNoW: A Winnow–based Classifier

SNoW stands for Sparse Network Of Winnows, and it is intended as a representative of on–line learning algorithms.

The basic component is the Winnow algorithm [123]. It consists of a linear threshold algorithm with multiplicative weight updating for 2-class problems, which learns very fast in the presence of many binary input features.

In the SNoW architecture there is a winnow node for each class, which learns to separate that class from all the rest. During training, each example is considered a positive example for winnow node associated to its class and a negative example for all the rest. A key point that allows a fast learning is that the winnow nodes are not connected to all features but only to those that are "relevant" for their class. When classifying a new example, SNoW is similar to a neural network which takes the input features and outputs the class with the highest activation.

A detailed description of the SNoW architecture and its current implementation is included in appendix A.1.

SNoW is proven to perform very well in high dimensional domains, where both, the training examples and the target function reside very sparsely in the feature space [188], e.g: text categorization, context–sensitive spelling correction, WSD, etc.

3.4.4 Boosting

The main idea of boosting algorithms is to combine many simple and moderately accurate hypotheses (called weak classifiers) into a single, highly accurate classifier. The weak classifiers are trained sequentially and, conceptually, each of them is trained on the examples which were the most difficult to classify by the preceding weak classifiers. These weak hypotheses are then linearly combined into a single rule called the combined hypothesis.

More particularly, the Schapire and Singer's AdaBoost.MH algorithm for multiclass multi–label classification [197] has been used. As in that paper, very simple weak hypotheses are used. They test the value of a boolean predicate and make a real–valued prediction based on that value. The predicates used, which are the binarization of the attributes described in section 3.5.2, are of the form "$f = v$", where f is a feature and v is a value (e.g: "previous_word = *hospital*"). Each weak rule uses a single feature, and, therefore, they can be seen as simple decision trees with one internal node (testing the value of a binary feature) and two leaves corresponding to the yes/no answers to that test.

A detailed description of Schapire and Singer's AdaBoost.MH algorithm for multiclass multi–label classification, and its implementation for WSD is included in appendix A.2.

LazyBoosting (LB)

LazyBoosting, explained in [72], is a simple modification of the AdaBoost.MH algorithm, which consists of reducing the feature space that is explored when learning each weak classifier. More specifically, a small proportion p of attributes are randomly selected and the best weak rule is selected only among them. The idea behind this method is that if the proportion p is not too small, probably a sufficiently good rule can be found at each iteration. Besides, the chance for a good rule to appear in the whole learning process is very high. Another important characteristic is that no attribute needs to be discarded and so we avoid the risk of eliminating relevant attributes. The method seems to work quite well since no important degradation is observed in performance for values of p greater or equal to 5% (this may indicate that there are many irrelevant or highly dependant attributes in our domain). Therefore, this modification significantly increases the efficiency of the learning process (empirically, up to 7 times faster) with no loss in accuracy.

3.5 Setting

3.5.1 The DSO Corpus

The DSO corpus is a semantically annotated corpus containing 192,800 occurrences of 121 nouns and 70 verbs[12], corresponding to the most frequent and ambiguous English words. This corpus was collected by Ng and colleagues [159] and it is available from the Linguistic Data Consortium (LDC)[13].

The DSO corpus contains sentences from two different corpora, namely Wall Street Journal (WSJ) and Brown Corpus (BC). Therefore, it is easy to perform experiments about the portability of alternative systems by training on the WSJ part and testing on the BC part, or vice-versa. Hereinafter, the WSJ part of DSO will be referred to as corpus A, and the BC part to as corpus B. At a word level, we force the number of examples of corpus A and B be the same[14] in order to have symmetry and allow the comparison in both directions.

From these corpora, a group of 21 words which frequently appear in the WSD literature has been selected to perform the comparative experiments (each word is treated as a different classification problem). These words are 13 nouns (age, art, body, car, child, cost, head, interest, line, point, state, thing, work) and 8 verbs (become, fall, grow, lose, set, speak, strike, tell). Table 7 contains information about the number of examples, the number of senses, and the percentage of the most frequent sense (MFS) of these reference words, grouped by nouns, verbs, and all 21 words.

		examples			senses			MFS (%)		
corpus	PoS	min	max	avg	min	max	avg	min	max	avg
A	nouns	122	714	420	2	24	7.7	37.9	90.7	59.8
	verbs	101	741	369	4	13	8.9	20.8	81.6	49.3
	all	101	741	**401**	2	24	**8.1**	20.8	90.7	**56.1**
B	nouns	122	714	420	3	24	8.8	21.0	87.7	45.3
	verbs	101	741	369	4	14	11.4	28.0	71.7	46.3
	all	101	741	**401**	3	24	**9.8**	21.0	87.7	**45.6**

Table 7: Information about the set of 21 words of reference.

[12]These examples, consisting of the full sentence in which the ambiguous word appears, are tagged with a set of labels corresponding, with minor changes [160], to the senses of WordNet 1.5 [146].

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

[14]This is achieved by reducing the size of the largest corpus (by performing random elimination of examples) to the size of the smallest.

3.5.2 Attributes

Two kinds of information are used to perform disambiguation: **local** and **topical** context.

Let "$\ldots w_{-3}\ w_{-2}\ w_{-1}\ w\ w_{+1}\ w_{+2}\ w_{+3} \ldots$" be the context of consecutive words around the word w to be disambiguated, and $p_{\pm i}$ $(-3 \leq i \leq 3)$ be the part–of–speech tag of word $w_{\pm i}$[15]. Attributes referring to local context are the following 15:

$$p_{-3},\ p_{-2},\ p_{-1},\ p_{+1},\ p_{+2},\ p_{+3},\ w_{-1},\ w_{+1},\ (w_{-2}, w_{-1}),\ (w_{-1}, w_{+1}),\ (w_{+1}, w_{+2}),$$
$$(w_{-3}, w_{-2}, w_{-1}),\ (w_{-2}, w_{-1}, w_{+1}),\ (w_{-1}, w_{+1}, w_{+2}) \text{ and } (w_{+1}, w_{+2}, w_{+3})$$

where the last seven correspond to collocations of two and three consecutive words.

The topical context is formed by $c_1, \ldots, c_m$, which stand for the unordered set of open class words appearing in the sentence[16].

The four methods tested translate this information into features in different ways. SNoW and LB algorithms require binary features. Therefore, local context attributes have to be binarized in a preprocess, while the topical context attributes remain as binary tests about the presence/absence of a concrete word in the sentence. As a result the number of attributes is expanded to several thousands (from 1,764 to 9,900 depending on the particular word).

The binary representation of the topical–context attributes is not appropriate for NB and EB algorithms. 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 thousands of 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) used in EB in favour of that stored examples which have less 1's, that is, those corresponding to the shortest sentences.

In order to address this limitation we propose [73] 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 closed class 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 $V_j = \{w_{j_1}, w_{j_2}, \ldots, w_{j_m}\}$ is redefined as: $S(V_i, V_j) = \| V_i \cap V_j \|$, that is, equal to the number of words shared[17].

This approach implies that a test example is classified taking into account the information about the words it contains (*positive* information), but not the information about the words it does not contain.

On the other hand, NB have problems with the efficiency when classifying new examples, since a product of thousands of conditional probabilities (real values) must be done to classify a new example. In order to improve performance, we tested a variant of the Naive Bayes algorithm which does not take into account all attributes but only the conditional probabilities corresponding to the words that appear in the test examples (again positive information). In this way, the efficiency is increased with no loss in accuracy [73].

[15] In this experiment, the Decision-tree-based tagger by [138] has been used to annotate training examples with the part–of–speech labels.

[16] The already described set of attributes contains those attributes used in [159], with the exception of the morphology of the target word and the verb–object syntactic relation.

[17] This measure is usually known as the *matching coefficient* [28]. More complex similarity measures, e.g. Jaccard or Dice coefficients, have not been explored.

These variants of NB and EB algorithms are called PNB and PEB, standing for Positive EB and NB. They are fully tested in [73] and empirically proven to perform much better than other variants of handling available attributes.

3.5.3 Experimental Methodology

The comparison of algorithms has been performed in series of controlled experiments using exactly the same training and test sets. There are 7 combinations of training–test sets called: A+B—A+B, A+B–A, A+B–B, A–A, B–B, A–B, and B–A, respectively. In this notation, the training set is placed at the left hand side of symbol "–", while the test set is at the right hand side. For instance, A–B means that the training set is corpus A and the test set is corpus B. The symbol "+" stands for set union, therefore A+B–B means that the training set is A union B and the test set is B.

When comparing the performance of two algorithms, two different statistical tests of significance have been applied depending on the case. A–B and B–A combinations represent a single training–test experiment. In this cases, the McNemar's test of significance is used (with a confidence value of: $\chi^2_{1,0.95} = 3.842$), which is proven to be more robust than a simple test for the difference of two proportions.

In the other combinations, a 10-fold cross-validation was performed in order to prevent testing on the same material used for training. In these cases, accuracy/error rate figures reported in following sections are averaged over the results of the 10 folds. The associated statistical tests of significance is a paired Student's t-test with a confidence value of: $t_{9,0.975} = 2.262$.

Information about both statistical tests can be found at [66].

When classifying test examples, all methods resolve potential ties between senses by choosing the most frequent sense among all those tied.

3.6 First Experiment

Table 8 shows the accuracy figures of the four methods in all combinations of training and test sets[18]. MFC stands for a Most–Frequent–sense Classifier, that is, a naive classifier that learns the most frequent sense of the training set and uses it to classify all examples of the test set. Averaged results are presented for nouns, verbs, and overall, and the best results for each case are printed in boldface.

It can be observed that LB outperforms all other methods in all cases. Additionally, this superiority is statistically significant, except when comparing LB to the EB approach in the cases marked with an asterisk. Note that, surprisingly, LB in A+B–A (or A+B–B) does not achieve substantial improvement to the results of A–A (or B–B) —in fact, the first variation is not statistically significant and the second is only slightly significant. That is, the knowledge acquired from a single corpus almost covers the knowledge of combining both corpora. This effect is also observed in the other methods, specially in some cases (e.g. SNoW in A+B–A vs. A–A) in which the joining of both training corpora is even counterproductive.

Regarding the portability of the systems, very disappointing results are obtained. Restricting to LB results, we observe that the accuracy obtained in A–B is 47.1% while the accuracy in B–B (which can be considered an upper bound for LB in B corpus) is 59.0%, that is, a difference of 12 points. Furthermore, 47.1% is only slightly better than the most frequent

[18]The second and third column correspond to the same train and test sets presented in [159, 156]

		Accuracy (%)						
Method	PoS	A+B—A+B	A+B–A	A+B–B	A–A	B–B	A–B	B–A
MFC	nouns	46.59	56.68	36.49	59.77	45.28	33.97	39.46
	verbs	46.49	48.74	44.23	48.85	45.96	40.91	37.31
	total	46.55	53.90	39.21	55.94	45.52	36.40	38.71
NB	nouns	62.29	68.89	55.69	66.93	56.17	36.62	45.99
	verbs	60.18	64.21	56.14	63.87	57.97	50.20	50.75
	total	61.55	67.25	55.85	65.86	56.80	41.38	47.66
EB	nouns	62.66	69.45	56.09	69.38	56.17	42.15	50.53
	verbs	63.67	68.39	58.58	68.25	59.57	51.19	52.24
	total	63.01	69.08	56.97	68.98	57.36	45.32	51.13
SNoW	nouns	61.24	66.36	56.11	68.85	56.55	42.13	49.96
	verbs	60.35	64.11	56.58	63.91	55.36	47.66	49.39
	total	60.92	65.57	56.28	67.12	56.13	44.07	49.76
LB	nouns	**66.00**	**72.09**	**59.92**	**71.69**	**58.33**	**43.92**	**51.28***
	verbs	**66.91**	**71.23**	**62.58**	**70.45***	**60.14***	**52.99**	**53.29***
	total	**66.32**	**71.79**	**60.85**	**71.26**	**58.96**	**47.10**	**51.99***

Table 8: Accuracy results of the methods on all training–test combinations

sense in corpus B, 45.5%. The comparison in the reverse direction is even worse: a difference of 19% from A–A to B–A, which is lower than the most frequent sense of corpus A, 55.9%.

Apart from accuracy figures, the observation of the predictions made by the four methods on the test sets provides interesting information about the comparison of the algorithms. Figure 1 shows the agreement rates and the Kappa (κ) statistics[19] between all pairs of methods in the A–A, A–B, B–B and B–A cases. 'DSO' stands for the annotation of DSO corpus, which is taken as the correct. Therefore the agreement rate with DSO contain the accuracy results previously reported. Some interesting conclusions can be drawn from those tables:

1. NB obtains the most similar results with regard to MFC in agreement rate and Kappa values in all tables. The average agreement ratio between the four tables is 83% and it achieves 89% in one case. This means that more than 8 out of 10 times it predicts the most frequent sense.

2. LB obtains the most similar results with regard to DSO (accuracy) in agreement rate and Kappa values, and it has the less similar Kappa and agreement values with regard to MFC (furthermore, LB provides the most dissimilar annotation with respect to the rest of algorithms). This indicates that LB is the method that better learns the behaviour of the DSO examples.

3. The Kappa values are very low. But, as it is suggested in [228], evaluation measures, such as precision and recall, should be computed relative to the agreement between the human annotators of the corpus and not to a theoretical 100%. It seems pointless to expect more agreement between the system and the reference corpus than between the

[19] The Kappa statistic k [48] is a better measure of inter–annotator agreement which reduces the effect of chance agreement. It has been used for measuring inter–annotator agreement during the construction of some semantic annotated corpora [228, 160].

annotators themselves. Contrary to the intuition that the agreement between human annotators should be very high in the WSD task, some papers report surprisingly low figures. For instance, [160] reports an accuracy rate of 56.7% and a Kappa value of 0.317 when comparing the annotation of a subset of the DSO corpus performed by two independent research groups[20]. Similarly, [228] reports values of Kappa near to zero when annotating some special words for the ROMANSEVAL corpus[21]. From this point of view, the Kappa values of 0.44 and 0.32 achieved by LB in A–A and B–B could be considered excellent results. Unfortunately, the subset of the DSO corpus and that used in this report are not the same and, therefore, a direct comparison is not possible. Furthermore, note that the LB Kappa values when moving from one corpus to another (A–B and B–A) are extremely low (0.14 and 0.18, respectively).

3.7 Second Experiment

The previous experiment shows that classifiers trained on the A corpus do not work well on the B corpus, and vice-versa. Therefore, it seems that a tuning process is necessary to make supervised systems portable.

This second experiment explores the effect of a tuning process consisting of adding to the original training set a relatively small sample of manually sense tagged examples of the new domain. The size of this supervised portion varies from 10% to 50% of the available corpus in steps of 10% (the remaining 50% is kept for testing). This set of experiments will be referred to as A+%B–B, or conversely, to B+%A–A.

In order to determine to which extent the original training set contributes to accurately disambiguate in the new domain, we also calculate the results for %A–A (and %B–B), that is, using only the tuning corpus for training.

Figure 2 graphically presents the results obtained by all methods. Each plot contains the X+%Y–Y and %Y–Y curves, and the straight lines corresponding to the lower bound MFC, and to the upper bounds Y–Y and X+Y–Y.

As expected, the accuracy of all methods grows (towards the upper bound) as more tuning corpus is added to the training set. However, the relation between X+%Y–Y and %Y–Y reveals some interesting facts. In plots 2a, 3a, and 1b the contribution of the original training corpus is null. Furthermore, in plots 1a, 2b, and 3b a degradation on the accuracy performance is observed. Summarizing, these six plots show that for Naive Bayes, Exemplar Based, and SNoW methods it is not worth keeping the original training examples. Instead, a better (but disappointing) strategy would be simply using the tuning corpus.

However, this is not the situation of LazyBoosting (plots 4a and 4b), for which a moderate (but consistent) improvement of accuracy is observed when retaining the original training set. Therefore, LazyBoosting shows again a better behaviour than their competitors.

3.8 Third Experiment

The bad results about the portability of systems could be explained by, at least, two reasons: 1) Corpus A and B have a very different distribution of senses, and, therefore, different a–

[20] A Kappa value of 1 indicates perfect agreement, while 0.8 is considered as indicating good agreement [40].

[21] ROMANSEVAL is, like SENSEVAL for English, a specific competition between WSD systems for Romance languages. See http://www.lpl.univ-aix.fr/projects/romanseval and http://www.itri.bton.ac.uk/events/senseval.

A–A

	DSO	MFC	NB	EB	SNoW	LB
DSO	—	0.57	0.67	0.70	0.69	0.72
MFC	-0.17	—	0.84	0.68	0.73	0.67
NB	0.19	0.02	—	0.80	0.85	0.77
EB	0.37	-0.04	0.38	—	0.79	0.79
SNoW	0.32	-0.06	0.51	0.48	—	0.77
LB	0.44	-0.14	0.32	0.48	0.43	—

A–B

	DSO	MFC	NB	EB	SNoW	LB
DSO	—	0.37	0.41	0.45	0.44	0.47
MFC	-0.26	—	0.89	0.64	0.71	0.62
NB	-0.12	0.04	—	0.70	0.78	0.68
EB	0.10	-0.11	0.16	—	0.70	0.70
SNoW	0.06	-0.08	0.24	0.34	—	0.68
LB	0.14	-0.11	0.14	0.39	0.33	—

B–B

	DSO	MFC	NB	EB	SNoW	LB
DSO	—	0.47	0.57	0.58	0.57	0.59
MFC	-0.20	—	0.78	0.65	0.62	0.57
NB	0.15	-0.04	—	0.77	0.77	0.70
EB	0.24	-0.14	0.40	—	0.71	0.71
SNoW	0.24	-0.15	0.41	0.37	—	0.69
LB	0.32	-0.17	0.30	0.43	0.39	—

B–A

	DSO	MFC	NB	EB	SNoW	LB
DSO	—	0.38	0.46	0.50	0.49	0.51
MFC	-0.32	—	0.81	0.62	0.56	0.52
NB	-0.05	-0.00	—	0.71	0.69	0.63
EB	0.11	-0.18	0.18	—	0.65	0.66
SNoW	0.13	-0.21	0.19	0.24	—	0.64
LB	0.18	-0.22	0.11	0.32	0.30	—

Figure 1: Kappa (κ) statistic (below diagonal) and agreement rate (above diagonal) between all methods in A–A, A–B, B–B and B–A experiments

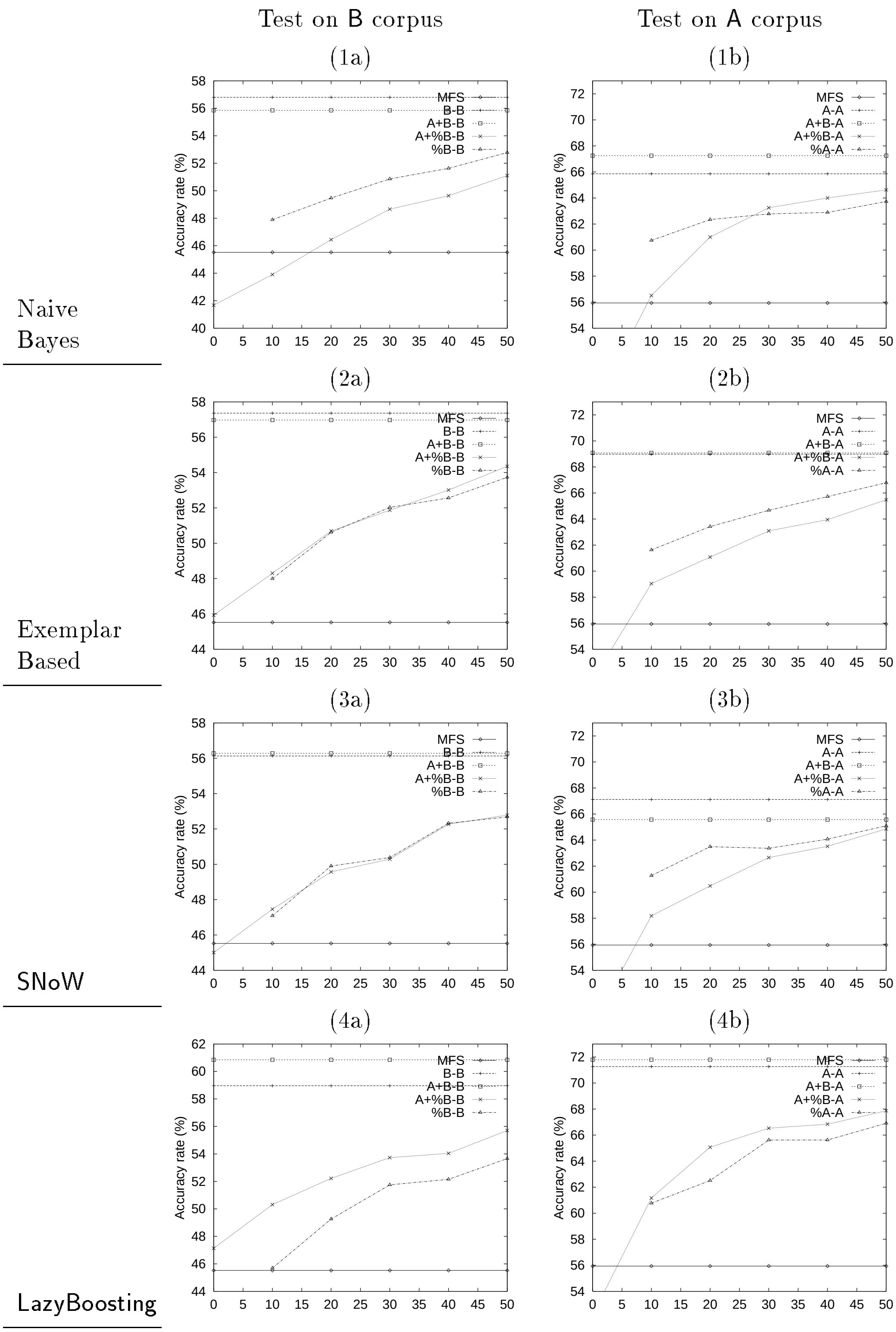

Figure 2: Results of the tuning experiment

priori biases; 2) Examples of corpus A and B contain different information, and, therefore, the learning algorithms acquire different (and non interchangeable) classification cues from both corpora.

The first hypothesis is confirmed by observing the bar plots of figure 3, which contain the distribution of the four most frequent senses of some sample words in the corpora A and B, respectively. As an example, first and second senses of noun *head* have a completely opposite distribution in both corpora A and B (first plot of figure 3).

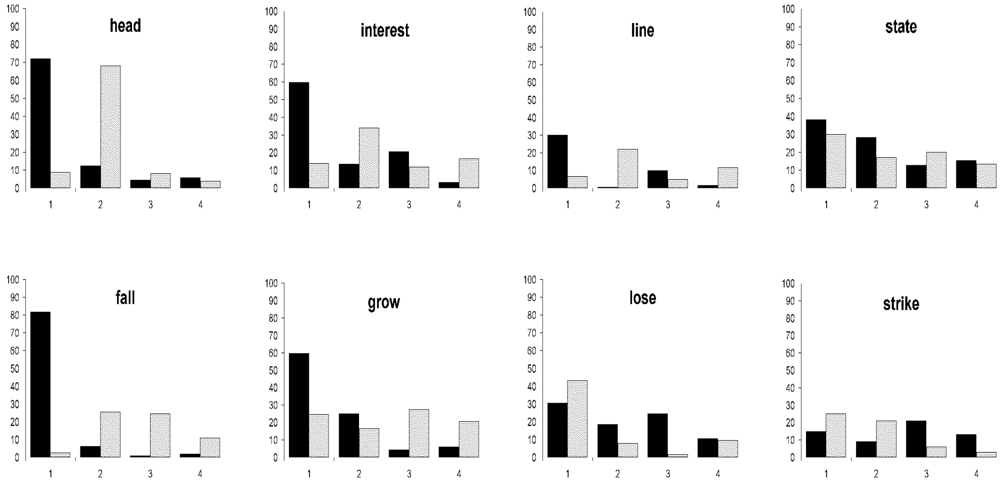

Figure 3: Distribution of the four most frequent senses for four nouns (head, interest, line, state) and four verbs (fall, grow, lose, strike). Black bars = A corpus; Grey bars = B corpus

In order to check the second hypothesis, two new sense–balanced corpora have been generated from the DSO corpus, by equilibrating the number of examples of each sense between A and B parts[22]. In this way, the first difficulty is artificially eliminated and the algorithms should be portable if examples of both parts were quite similar.

Table 9 shows the results obtained by LazyBoosting on these new corpora.

		Accuracy (%)						
Method	PoS	A+B—A+B	A+B–A	A+B–B	A–A	B–B	A–B	B–A
MFC	nouns	48.75	48.90	48.61	48.87	48.61	48.99	48.99
	verbs	48.22	48.22	48.22	48.22	48.22	48.22	48.22
	total	48.55	48.64	48.46	48.62	48.46	48.70	48.70
LB	nouns	62.82	64.26	61.38	63.19	60.65	53.45	55.27
	verbs	66.82	69.33	64.32	68.51	63.49	60.44	62.55
	total	64.35	66.20	62.50	65.22	61.74	56.12	58.05

Table 9: Results of LazyBoosting on the sense–balanced corpora

Regarding portability, we observe a significant accuracy decrease of 7 and 5 points from A–A to B–A, and from B–B to A–B, respectively[23]. That is, even when the same distribution

[22] Again, this is achieved by reducing the size of the largest corpus to the size of the smallest.

[23] This difference in accuracy is not as important as in the first experiment, due to the simplification provided by the balancing of sense distributions.

of senses is conserved between training and test examples, the portability of the supervised WSD systems is not guaranteed. As an extreme example, table 10 shows the performance of LB on the noun *state*. Note that in this case, LB performs even worse than MFC in the A–B and B–A columns.

	Accuracy (%)						
Method	A+B—A+B	A+B–A	A+B–B	A–A	B–B	A–B	B–A
MFC	39.00	39.00	39.00	39.00	39.00	39.00	39.00
LB	51.25	49.00	53.50	45.50	50.50	37.83	38.50

Table 10: Results of LazyBoosting on noun *state* (from the sense–balanced corpora).

These results imply that examples have to be largely different from one corpus to another. After the training process, LB provides as output an ordered set of rules based on the most relevant attributes. By studying the weak rules generated by LB from both corpora, we could observe the difference between them.

- The type of features used in the rules were significantly different between corpora, and, additionally, there were very few rules that apply to both sets. For corpus B, topical rules (i.e. those referring to topical context) seem to be more important than for corpus A. In the 50 firsts rules, corpus B has 30 topical rules and corpus A 20. Restricting to the 20 firsts rules, corpus B has 10 topical rules while corpus A has only 3. Not surprisingly, there are 34 of the 50 firsts rules learned from corpus A that do not appear in those learned from corpus B.
- The sign of the predictions of many of these common rules was somewhat contradictory between corpora. See, for instance, the two rules presented in table 11, acquired from corpus A and B respectively, which account for the collocation *state court*. For each rule and each sense there are two real numbers, which are the rule outputs when the feature holds in the example and when it does not, respectively. The sign of real numbers is interpreted as a prediction as to whether the sense should or should not be assigned to the example. The magnitude of the prediction is interpreted as a measure of confidence in the prediction. Therefore, according to the first rule, the presence of the collocation *state court* gives positive evidence to the sense number 3 and negative to the rest of senses. On the contrary, according to the second rule the presence of such collocation would contribute to sense number 5 and negatively to all the rest (including sense number 3).

			Senses					
		Feature	0	1	2	3	4	5
rule 1	A	*state court*	-0.5064	-1.0970	-0.6992	**1.2580**	-1.0125	-0.4721
			0.0035	0.0118	0.0066	-0.0421	0.0129	0.0070
rule 2	B	*state court*	-0.0696	-0.2988	-0.1476	-1.1580	-0.5095	**1.2326**
			-0.0213	0.0019	-0.0037	0.0102	0.0027	-0.0094

Table 11: Example of WeakRule: it is referred to the collocation *state court*.

It has to be noted that these rules are completely coherent with the senses assigned to the examples containing *state court* in both corpora, and therefore, the contradiction comes from the different information that characterize examples of both corpora.

3.9 Conclusions

This work has pointed out some difficulties regarding the portability of supervised WSD systems, a very important issue that has been paid little attention up to the present. The main conclusion that can be extracted is that to assure the portability of systems, a process of tuning to the new domain is required (at least if the learning–testing corpora differ so as BC and WSJ do). This result is in contradiction with the idea of "robust broad-coverage WSD" introduced by [158], in which a supervised system trained on a large enough corpora (say a thousand examples per word) should provide accurate disambiguation on any corpora (or, at least significantly better than MFS). Consequently, it is our belief that a number of issues regarding portability, tuning, knowledge acquisition, etc., should now be thoroughly studied before stating that the supervised ML paradigm is able to resolve a realistic WSD problem.

Regarding the ML algorithms tested, the contribution of this work consist of empirically demonstrating that the LazyBoosting algorithm outperforms other three state-of-the-art supervised ML methods for WSD. Furthermore, this algorithm is proven to have better properties when is applied to new domains.

Further work is planned to be done in the following directions:

- Studying the problem of obtaining a representative enough training corpus, clarifying what is exactly meant by *enough* and *representative*. Note that in current approaches to WSD each word is treated as a different classification problem. Therefore, the collection of a representative corpus should take into account the particularities of each word in order to decide the number of examples needed to learn, the most useful attributes, and so on.

- Due to the fact that most of the knowledge learned from a domain is not applicable when changing to a new domain, further investigation is needed on tuning strategies, specially on those using non-supervised algorithms.

- Extensively evaluate LazyBoosting on the WSD task. This would include taking into account additional/alternative attributes and testing the algorithm in other corpora —specially on sense-tagged corpora automatically obtained from Internet using non-supervised methods [112, 145].

- It is known that mislabelled examples resulting from annotation errors tend to be hard examples to classify correctly, and, therefore, tend to have large weights in the final distribution. This observation allows both to identify the noisy examples and use LazyBoosting as a way to improve data quality. Preliminary experiments have been already carried out in this direction on the DSO corpus.

- The inspection of the rules learned by LazyBoosting could provide evidence about similar behaviours of a–priori different senses. We think that this type of knowledge could be useful to perform clustering of too fine-grained or artificial senses.

4 Acknowledgements

This research has been partially funded by the Spanish Research Department (CICYT's BASURDE 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).

References

[1] S. Abney. *Parsing by Chunks.* R. Berwick, S. Abney and C. Tenny (eds.) Principle–based Parsing . Kluwer Academic Publishers, Dordrecht, 1991. http://www.sfs.nphil.uni-tuebingen.de/∼abney.

[2] S. Abney, R. E. Schapire, and Y. Singer. Boosting Applied to Tagging and PP–attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[3] D. Aha, D. Kibler, and M. Albert. Instance–based Learning Algorithms. *Machine Learning*, 7:37–66, 1991.

[4] David Aha. *Lazy Learning.* Kluwer Academic Publishers, Dordrecht, 1997. Reprinted from: Artificial Intelligence Review, 11:1–5.

[5] C. Aone and W. Bennett. Evaluating Automated and Manual Acquisition of Anaphora Resolution. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[6] C. Aone and K. Hausman. Unsupervised Learning of a Rule-based Spanish Part–of–speech Tagger. In *Proceedings of the 16th International Conference on Computatioal Linguistics, COLING*, pages 53–58, August 1996.

[7] S. Argamon, I. Dagan, and Y. Krymolowski. A Memory–based Approach to Learning Shallow Natural Language Patterns. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 67–73, Montréal, Canada, 1998. cmp–lg/9806011.

[8] L. R. Bahl, P. F. Brown, P. V. DeSouza, and R. L. Mercer. A Tree-Based Statistical Language Model for Natural Language Speech Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(7):1001–1008, 1989.

[9] G. Bakiri and T. G. Dietterich. Achieving High–Accuracy Text–to–Speech with Machine Learning. In B. Damper, editor, *Data Mining in Speech Synthesis.* Chapman and Hall, To appear in 1999.

[10] A. Berger, S. Della Pietra, and V. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.

[11] D. Bikel, R. Schwartz, and R. Weischedel. An Algorithm that Learns What's in a Name. *Machine Learning*, 34:211–231, 1999.

[12] E. Black, F. Jelinek, J. Lafferty, D. Magerman, R. L. Mercer, and S. Roukos. Towards History–based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, San Mateo, CA, 1992. cmp-lg/9405007.

[13] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co–Training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory, COLT-98*, pages 92–100, Madison, Wisconsin, 1998.

[14] A. van den Bosch, W. Daelemans, and T. Weijters. Morphological Analysis as Classification: an Inductive–Learning Approach. In *Proceedings of the 2nd NeMLaP*, 1996. cmp-lg/9607021.

[15] A. van den Bosch, T. Weijters, and W. Daelemans. Modularity in Inductively–learned Word Pronunciation Systems. In *Proceedings of the NeMLaP-3/CoNLL'98*, pages 185–194, 1998. cmp-lg/9801004.

[16] T. Brants, W. Skut, and B. Krenn. Tagging Grammatical Functions. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*, Brown University, Providence, RI, 1997. cmp-lg/9707015.

[17] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. The Wadsworth Statistics/Probability Series. Wadsworth International Group, Belmont, CA, 1984.

[18] E. Brill. Automatic Grammar Induction and Parsing Free Text: A Transformation–based Approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993. http://www.cs.jhu.edu/~brill/acadpubs.html.

[19] E. Brill. Transformation–based Error–driven Learning and Natural Language Processing: A Case Study in Part–of–speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[20] E. Brill and P. Resnik. A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING*, Kyoto, Japan, August 1994. http://www.cs.jhu.edu/~brill/acadpubs.html.

[21] Eric Brill. A Simple Rule–Based Part–of–speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, pages 152–155. ACL, 1992.

[22] Eric Brill. Some Advances in Rule–based Part–of–speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI*, pages 722–727, 1994. http://www.cs.jhu.edu/~brill/acadpubs.html.

[23] Eric Brill. Unsupervised Learning of Disambiguation Rules for Part–of–speech Tagging. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 1–13, Massachusetts, 1995. http://www.cs.jhu.edu/~brill/acadpubs.html.

[24] Eric Brill and Jun Wu. Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the joint 17th International Conference on Computational Linguistics and*

36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL, pages 191–195, Montréal, Canada, 1998. http://www.cs.jhu.edu/~brill/acadpubs.html.

[25] E. J. Briscoe. *Prospects for Practical Parsing of Unrestricted Text: Robust Statistical Parsing Techniques.* N. Oostdijk and P. de Haan (eds.), Corpus-Based Research into Language. Rodopi, Amsterdam, 1994.

[26] P. F. Brown, S. Della Pietra, V. Della Pietra, and R. L. Mercer. Word Sense Disambiguation using Statistical Methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 264–270, 1991.

[27] R. F. Bruce and J. M. Wiebe. Decomposable Modeling in Natural Language Processing. *Computational Linguistics*, 25(2):195–207, 1999.

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

[29] M. E. Califf and R. J. Mooney. Applying ILP-based Techniques to Natural Language Information Extraction: An Experiment in Relational Learning. In *Workshop Notes of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming*, pages 7–11, Nagoya, Japan, 1997.

[30] M. E. Califf and R. J. Mooney. Relational Learning of Pattern–match Rules for Information Extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, FL, 1999.

[31] C. Cardie. Embedded Machine Learning Systems for Natural Language Processing: A General Framework. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[32] C. Cardie and R. Mooney. Guest Editors' Introduction: Machine Learning and Natural Language. *Machine Learning. Special Issue on Natural Language Learning*, 34(1/2/3):5–9, 1999.

[33] C. Cardie and D. Pierce. Error–driven Pruning of Treebank Grammars for Base Noun Phrase Identification. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, Montréal, Canada, 1998.

[34] C. Cardie and K. Wagstaff. Noun Phrase Coreference as Clustering. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC*, 1999.

[35] Claire Cardie. Learning to Disambiguate Relative Pronouns. In *Proceedings of the 10th National Conference on Artificial Intelligence, AAAI*, pages 38–43, San Jose, CA, 1992. AAAI Press / MIT Press.

[36] Claire Cardie. A Case–based Approach to Knowledge Acquisition for Domain–specific Sentence Analysis. In *Proceedings of the 11th National Conference on Artificial Intelligence, AAAI*, pages 798–803, 1993. AAAI Press / MIT Press.

[37] Claire Cardie. Using Decision Trees to Improve Case–based Learning. In *Proceedings of the 10th International Conference on Machine Learning, ICML'93*, pages 25–32, Amherst, MA, 1993. Morgan Kaufmann.

[38] Claire Cardie. *Domain Specific Knowledge Acquisition for Conceptual Sentence Analysis.* Phd. Thesis, University of Massachusets, 1994. Available as University of Massachusetts, CMPSCI Technical Report 94-74.

[39] Claire Cardie. Automatic Feature Set Selection for Case-Based Learning of Linguistic Knowledge. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 113–126, 1996.

[40] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[41] E. Charniak. *Statistical Language Learning.* The MIT Press, Cambridge, Massachusetts, 1993.

[42] E. Charniak. Statistical Techniques for Natural Language Parsing. *AI Magazine*, 1997. http://www.cs.brown.edu/people/ec.

[43] T. Chen, V. Soo, and A. Lin. Learning to Parse with Recurrent Neural Networks. In *Proceedings of European Conference on Machine Learning Workshop on Machine Learning and Text Analysis, ECML*, pages 63–68, 1993.

[44] R. Christensen. *Log-Linear Models and Logistic Regression.* Springer Texts in Statistics, G. Casella, S. Fienberg and I. Olkin (eds.). Springer, 1997.

[45] K. W. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the 1st Conference on Applied Natural Language Processing, ANLP*, pages 136–143. ACL, 1988.

[46] K. W. Church and R. L. Mercer. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics*, 19(1), 1993.

[47] P. Clark and T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3:261–284, 1989.

[48] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Journal of Educational and Psychological Measurement*, 20:37–46, 1960.

[49] W. Cohen. Text Categorization and Relational Learning. In *Proceedings of the 12th International Conference on Machine Learning*, pages 124–132, San Francisco, CA, 1995. Morgan Kaufmann.

[50] W. Cohen. Learning to Classify English Text with ILP Methods. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 124–143. IOS Press, Amsterdam, 1996.

[51] W. Cohen and Y. Singer. Context–sensitive Learning Methods for Text Categorization. In *Proceedings of the 19th Annual Inter. ACM Conference on Research and Development in Information Retrieval*, 1996.

[52] M. Collins and J. Brooks. Prepositional phrase Attachment Trhough a Backed–off Model. In *Proceedings of the 3rd Workshop on Very Large Corpora*, Massachusetts, 1995.

[53] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 509–516, 1998.

[54] D. Cutting, J. Kupiec, J. Pederson, and P. Sibun. A Practical Part–of–speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, pages 133–140. ACL, 1992.

[55] W. Daelemans. *Memory–based Lexical Acquisition and Processing.* Machine Translation and the Lexicon, Lecture Notes in Artificial Intelligence 898. P. Steffens editor. Springer, Berlin, 1995.

[56] W. Daelemans, P. Berck, and S. Gillis. Unsupervised Discovery of Phonological Categories through Supervised Learning of Morphological Rules. In *Proceedings of 16th International Conference on Computational Linguistics, COLING*, pages 95–100, Copenhagen, Denmark, 1996.

[57] W. Daelemans, A. van den Bosch, and T. Weijters. *IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms.* D. Aha (ed.), Artificial Intelligence Review 11, Special issue on Lazy Learning. Kluwer Academic Publishers, 1997.

[58] W. Daelemans, A. van den Bosch, and J. Zavrel. Forgetting Exceptions is Harmful in Language Learning. *Machine Learning*, 34:11–41, 1999.

[59] W. Daelemans, T. Weijters, and A. van den Bosch. *Empirical Learning of Natural Language Processing Tasks.* Lecture Notes in Artificial Intelligence, number 1224. Springer–Verlag, Berlin, 1997.

[60] W. Daelemans, J. Zavrel, and P. Berck. Part-of-Speech Tagging for Dutch with MBT, a Memory-based Tagger Generator. In *Congresboek van de Interdisciplinaire Onderzoeksconferentie Informatiewetenchap*, TU Delft, 1996.

[61] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. MBT: A Memory–Based Part–of–speech Tagger Generator. In *Proceedings of the 4th Workshop on Very Large Corpora*, pages 14–27, Copenhagen, Denmark, 1996.

[62] I. Dagan, Y. Karov, and D. Roth. Mistake-Driven Learning in Text Categorization. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*, Brown University, Providence, RI, 1997.

[63] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society*, 39, 1977.

[64] A. Díaz, M. de Buenaga, L. A. Ureña, and M. García. Integrating Linguistic Resources in a Uniform Way for Text Classification Tasks. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 1197–1204, Granada, Spain, May 1998.

[65] T. G. Dietterich. Machine Learning Research: Four Current Directions. *AI Magazine*, 18(4):97–136, 1997.

[66] T. G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7), 1998.

[67] L. Dini, V. Di Tomaso, and F. Segond. Word Sense Disambiguation with Functional Relations. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 1189–1196, Granada, Spain, May 1998.

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

[69] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive Learning Algorithms and Representations for Text Categorization. In *CIKM-98: Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998.

[70] M. Eineborg and B. Gambäck. Tagging Experiments Using Neural Networks. In *Proceedings of the 9th Nordic Conference of Computational Linguistics*, Stockholm, Sweden, 1993.

[71] S. P. Engelson and I. Dagan. Minimizing Manual Annotation Cost in Supervised Training from Corpora. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[72] G. Escudero, L. Màrquez, and G. Rigau. Boosting Applied to Word Sense Disambiguation. In *To appear in Proceedings of the 12th European Conference on Machine Learning, ECML*, Barcelona, Spain, 2000.

[73] G. Escudero, L. Màrquez, and G. Rigau. Naive Bayes and Exemplar-Based Approaches to Word Sense Disambiguation Revisited. In *To appear in Proceedings of the 14th European Conference on Artificial Intelligence, ECAI*, Berlin, Germany, 2000.

[74] G. Escudero, L. Màrquez, and G. Rigau. On the Portability and Tuning of Supervised Word Sense Disambiguation Systems. Research Report LSI-00-30-R, Software Department (LSI). Technical University of Catalonia (UPC), Barcelona, Catalonia, 2000.

[75] S. Federici, S. Montemagni, V. Pirrelli, and N. Calzolari. Analogy–based Extraction of Lexical Knowledge from Corpora: The SPARKLE Experience. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 75–82, Granada, Spain, May 1998.

[76] S. Federici and V. Pirrelli. *Analogy, Computation and Linguistic Theory.* D. Jones editor, New Methods in Language Processing. London: UCL Press, 1996.

[77] D. Fisher, M. Pazzani, and P. Langley. *Concept Formation: Knowledge and Experience in Unsupervised Learning.* Morgan Kaufmann, 1991.

[78] D. H. Fisher. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, 2(2):139–172, 1991.

[79] D. Freitag. Information extraction from HTML: Application of a general learning approach. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 517–523, 1998.

[80] D. Freitag. Multi–strategy learning for information extraction. In *Proceedings of the 15th International Conference on Machine Learning*, pages 161–169, 1998.

[81] D. Freitag. Toward general–purpose learning for information extraction. In *Proceedings of the joint 17th International Conference on Computational Linguistics and the 36th Annual Meeting of Association for Computational Linguistics, COLING–ACL*, 1998.

[82] Y. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. In L. Saitta, editor, *Proceedings of the 13th International Conference on Machine Learning, ICML'96*, pages 148–156, San Francisco, CA, 1996. Morgan Kaufmann.

[83] Y. Freund and R. E. Schapire. A Decision–Theoretic Generalization of On–line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[84] A. Fujii, K. Inui, T. Tokunaga, and H. Tanaka. Selective Sampling for Example–based Word Sense Disambiguation. *Computational Linguistics*, 24(4):573–598, 1998.

[85] W. Gale, K. W. Church, and D. Yarowsky. One Sense Per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237, Arden House, Harriman, New York, US, 1992.

[86] 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.

[87] D. E. Goldberg, editor. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison–Wesley, Reading, Mass., 1989.

[88] A. R. Golding. A Bayesian–hybrid Method for Context–sensitive Spelling Correction. In *Proceedings of the 3rd Workshop on Very Large Corpora.* ACL, 1995.

[89] A. R. Golding and D. Roth. A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, 34:107–130, 1999.

[90] H. van Halteren, J. Zavrel, and W. Daelemans. Improving Data Driven Wordclass Tagging by System Combination. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 491–497, Montréal, Canada, August 1998.

[91] D. Harman. *Relevance Feedback and other Query Modification Techniques.* W. B. Frakes and R. Baeza–Yates (eds.), Information Retrieval: Data Structures and Algorithms. Prentice–Hall, Inc., Englewood Cliffs, NJ, 1992.

[92] M. Haruno, S. Shirai, and Y. Ooyama. Using Decision Trees to Construct a Practical Parser. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, Montréal, Canada, August 1998.

[93] M. Haruno, S. Shirai, and Y. Ooyama. Using Decision Trees to Construct a Practical Parser. *Machine Learning*, 34(1/2/3):131–151, 1999.

[94] S. Haykin. *Neural Networks.* Macmillan College Publishing Company, Inc., 1994.

[95] M. Hearst and D. D. Palmer. Adaptive Sentence Boundary Disambiguation. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, Stuttgart, Germany, October 1994. ACL.

[96] 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.

[97] F. Jelineck, R. Mercer, and S. Roukos. *Principles of Lexical Language Modeling for Speech Recognition.* S. Furni and M. M. Sondhi (eds.), Advances in Speech Processing. Marcel Dekker, Inc., New York, 1992.

[98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[99] T. Joachims. Transductive inference for text classification using support vector machines. In Ivan Bratko and Saso Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

[100] D. Jones. *Analogical Natural Language Processing.* London: UCL Press, 1996.

[101] S. Y. Jung, Y. C. Park, K. S. Choi, and Y. Kim. Markov Random Field Based English Part–of–speech Tagging System. In *Proceedings of 16th International Conference on Computational Linguistics, COLING*, Copenhagen, Denmark, August 1996.

[102] D. Jurafsky and J.H. Martin. *Speech and Language Processing.* Prentice Hall, To appear in 1999. http://www.cs.colorado.edu/∼martin/slp.html.

[103] A. Kempe. Probabilistic Tagging with Feature Structures. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING*, pages 161–165, Kyoto, Japan, August 1994. cmp–lg/9410027.

[104] A. Kilgarriff and J. Rosenzweig. English SENSEVAL: Report and Results. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC*, Athens, Greece, to appear.

[105] J. Kivinen and M. K. Warmuth. Exponentiated Gradient versus Gradient Descent for Linear Predictors. Technical Report UCSC–CRL–94–16, Basking Center for Computer Engineering and Information Sciences. University of California, Santa Cruz, CA, 1994.

[106] B. Krenn and C. Samuelsson. The Linguists' Guide to Statistics: Don't Panic. Technical report, Universität des Saarlandes, 1997. Postscript version of December 19, 1997 at URL: http://coli.uni-sb.de/∼christer.

[107] Y. Krymolowski and D. Roth. Incorporating Knowledge in Natural Language Learning: A Case Study. In *Proceedings of the COLING-ACL Workshop on the Usage of WordNet in Natural Language Processing Systems*, Montréal, Canada, August 1998. http://l2r.cs.uiuc.edu/~danr.

[108] M. M. Lankhorst. Automatic Word Categorization with Genetic Algorithms. Technical Report, Dept. of CS. University of Groningen, Groningen, The Netherlands, 1994.

[109] M. M. Lankhorst. Breeding Grammars. Grammatical Inference with a Genetic Algorithm. Technical Report, Dept. of CS. University of Groningen, Groningen, The Netherlands, 1994.

[110] R. Lau, R. Rosenfeld, and S. Roukos. Adaptive Language Modelling Using the Maximum Entropy Principle. In *Proceedings of Human Language Technology Workshop, ARPA*, 1993.

[111] S. Lawrence, F. Sandiway, and C. L. Giles. Natural Language Grammatical Inference: A Comparison of Recurrent Neural Networks and Machine Learning Methods. In *Proceedings of the IJCAI Workshop in New Approaches for NLP*, 1995. Also in S. Wermter, E. Riloff and G. Scheler (editors), Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing. Computer Notes in Artificial Intelligence 1040, Springer 1996.

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

[113] B. J. Lee. Applying Parallel Learning Models of Artificial Neural Networks to Letters Recognition from Phonemes. In P. Chan, editor, *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, pages 66–71, 1996. http://www.cs.fit.edu/~imlm.

[114] G. Leech and S. Fligelstone. *Computers and Corpus Analysis.* C. S. Butler, editor, Computers and Written Texts. Blackwell, Oxford UK & Cambridge USA, 1992.

[115] W. Lehnert. *Symbolic/subsymbolic Sentence Analysis: Exploiting the Best of two Worlds.* J. Barnden and J. Pollack, editors, Advances in Connectionist and Neural Computation. Ablex Publishers, Norwood, NJ, 1991.

[116] Mark Lewellen. Neural Network Recognition of Spelling Errors. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 1490–1493, Montréal, Canada, August 1998.

[117] D. Lewis and M. Ringuette. A Comparison of two Learning Algorithms for Text Categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994.

[118] D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka. Training Algorithms for Linear Text Classifiers. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval, SIGIR*, pages 298–306, 1996.

[119] D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec and C. Rouveirol, editor, *LNAI 1398: Proceedings of the 10th European Conference on Machine Learning, ECML'98*, pages 4–15, Chemnitz, Germany, 1998. Springer.

[120] M. Liberman and Y. Schabes. Statistical Methods in Natural Language Processing. Tutorial Notes, Computer and Information Science Department. University of Pennsylvania, 1993.

[121] R. P. Lippmann. Review of Neural Networks for Speech Recognition. *Neural Computation*, 1:1–38, 1989.

[122] D. J. Litman. Classifying Cue Phrases in Text and Speech Using Machine Learning. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI*, pages 806–813, 1994. AAAI Press / MIT Press.

[123] N. Littlestone. Learning Quickly when Irrelevant Attributes Abound. *Machine Learning*, 2:285–318, 1988.

[124] N. Littlestone and M. K. Warmuth. The Weighted Majority Algorithm. *Information and Computation*, 108(2):212–261, 1994.

[125] Joan López. *Un enfoque neuronal para la desambiguación del significado*. Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, 1998.

[126] R. López de Mántaras and E. Plaza. Case Based Reasoning: An Overview. *AI Communications*, 10:21–29, 1997.

[127] R. M. Losee. Learning Syntactic Rules and Tags with Genetic Algorithms for Information Retrieval and Filtering: An Empirical Basis for Grammatical Rules. *Information Processing & Management*, May 1994.

[128] C. Lyon. *The Representation of Natural Language to Enable Neural Networks to Detect Syntactic Structures*. Phd. Thesis, Computer Science Department, University of Hertfordshire, UK, 1994.

[129] C. Lyon and B. Dickerson. A Fast Partial Parse of Natural Language Sentences using a Connectionist Method. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, pages 149–156, Dublin, Ireland, 1995.

[130] M. Muñoz and V. Punyakanok and D. Roth and D. Zimak. A Learning Approach to Shallow Parsing. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC*, 1999.

[131] Qing Ma and Hitoshi Isahara. A Multi-Neuro Tagger Using Variable Lengths of Contexts. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 802–806, Montréal, Canada, August 1998.

[132] D. M. Magerman. Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. ACL, 1995.

[133] L. Mangu and E. Brill. Automatic Rule Acquisition for Spelling Correction. In *Proceedings of the 14th International Conference on Machine Learning, ICML'97*, pages 734–741, 1997. http://www.cs.jhu.edu/~brill/acadpubs.html.

[134] I. Mani and E. Bloedorn. Machine Learning of Generic and User-Focused Summarization. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI/IAAI*, pages 821–826. AAAI Press / The MIT Press, 1998.

[135] N. M. Marques, G. P. Lopes, and C. A. Coelho. Learning Verbal Transitivity Using LogLinear Models. In C. Nédellec and C. Rouveirol, editor, *LNAI 1398: Proceedings of the 10th European Conference on Machine Learning, ECML'98*, pages 19–24, Chemnitz, Germany, 1998. Springer.

[136] L. Màrquez. *Part–of–Speech Tagging: A Machine–Learning Approach based on Decision Trees*. Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, 1999.

[137] L. Màrquez, L. Padró, and Horacio Rodríguez. A Machine Learning Approach to POS Tagging. *Machine Learning Journal*, 39(1), 2000.

[138] L. Màrquez, H. Rodríguez, J. Carmona, and J. Montolio. Improving POS Tagging Using Machine Learning Techniques. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC*, 1999.

[139] Lluís Màrquez, Lluís Padró, and Horacio Rodríguez. Improving Tagging Accuracy by Voting Taggers. In *Proceedings of the 2nd Conference on Natural Language Processing & Industrial Applications, NLP+IA/TAL+AI*, pages 149–155, New Brunswick, Canada, August 1998.

[140] Lluís Màrquez and Horacio Rodríguez. Towards Learning a Constraint Grammar from Annotated Corpora Using Decision Trees. Working Paper #21, ESPRIT BRA–7315 Acquilex II, 1995.

[141] Lluís Màrquez and Horacio Rodríguez. Automatically Acquiring a Language Model for POS Tagging Using Decision Trees. In *Proceedings of the Second Conference on Recent Advances in Natural Language Processing, RANLP*, pages 27–34, Tzigov Chark, Bulgaria, September 1997.

[142] A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*. AAAI Press, 1998. http://www.cs.cmu.edu/~mccallum.

[143] J. F. McCarthy and W. G. Lehnert. Using Decision Trees for Coreference Resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI*, pages 1050–1055, 1995.

[144] B. Merialdo. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171, 1994.

[145] 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.

[146] 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.

[147] R. Mitkov. Robust Pronoun Resolution with Limited Knowledge. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 869–875, Montréal, Canada, 1998.

[148] R. Mitkov, L. Belguith, and M. Stys. Multilingual Robust Anaphora Resolution. In *Proceedings of the 3rd International Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 7–16, Granada, Spain, 1998.

[149] R. J. Mooney. Encouraging Experimental Results on Learning CNF. *Machine Learning*, 19(1):79–92, 1995.

[150] 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.

[151] R. J. Mooney. *Inductive Logic Programming for Natural Language Processing.* S. Muggleton (Ed.), Inductive Logic Programming: Selected Papers from the 6th International Workshop. Springer Verlag, Berlin, 1997.

[152] R. J. Mooney and M. E. Califf. Induction of First–order Decision Lists: Results on Learning the Past Tense of English Verbs. *Journal of Artificial Intelligence Research*, 3:1–24, 1995.

[153] R. J. Mooney and M. E. Califf. Learning the Past Tense of English Verbs Using Inductive Logic Programming. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[154] S. K. Murthy. *On Growing Better Decision Trees from Data.* Phd. Thesis, Johns Hopkins University, Baltimore, Maryland, 1995.

[155] M. Nakamura, K. Maruyama, T. Kawabata, and K. Shikano. Neural Network Approach to Word Category Prediction for English Texts. In *Proceedings of 13th International Conference on Computational Linguistics, COLING*, pages 213–218, Helsinki, Finland, 1990. Karlgren, H (ed.) COLING 90.

[156] 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.

[157] 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.

[158] 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?*, Washington, USA, 1997.

[159] 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.

[160] H. T. Ng, C. Y. Lim, and S. K. Foo. A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. In *Proceedings of the ACL SIGLEX Workshop: Standardizing Lexical Resources*, College Park, MD, USA, 1999.

[161] K. Nigam, J. Lafferty, and A. McCallum. Using Maximum Entropy for Text Classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61-67. 1999.*, pages 61–67, 1999.

[162] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to Classify Text from Labeled and Unlabeled Documents. In *Proceedings of the 15th National Conference on Artificial Intelligence, AAAI-98*, Madison, Wisconsin, 1998.

[163] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning Journal, to appear.* http://www.cs.cmu.edu/~knigam/.

[164] G. Orphanos and D. Christodoulakis. Pos disambiguation and unkown word guessing with decision trees. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, Bergen, Norway, 1999.

[165] T. Pedersen. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. 2000. `arXiv:cs.CL/0005006 - 7 May 2000`.

[166] T. Pedersen and R. Bruce. Knowledge Lean Word-Sense Disambiguation. In *Proceedings of the 15th National Conference on Artificial Intelligence*. AAAI Press, 1998.

[167] F. Pereira and Y. Schabes. Inside-Outside Re-estimation from Partially Bracketed Corpora. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 128–135, 1992.

[168] S. Della Pietra, V. Della Pietra, and John Lafferty. Inducing Features of Random Fields. Technical Report CMU–CS95–144, School of Computer Science, Carnegie–Mellon University, 1995.

[169] D. M. Powers. Machine Learning of Natural Language. In *Joint ACL/EACL Tutorial Program*, Madrid, Spain, 1997.

[170] J. R. Quinlan. *Discovering Rules from Large Collections of Examples*. Edimburgh University Press, 1979. ???

[171] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[172] J. R. Quinlan. Learning Logical Definitions from Relations. *Machine Learning*, 5(3):239–266, 1990.

[173] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993.

[174] J. R. Quinlan. Boosting First–Order Learning. In *Proceedings of the ALT'96 conference*, 1996. http://www.cse.unsw.EDU.AU/~quinlan.

[175] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Readings in Speech Recognition (eds. A. Waibel, K. F. Lee). Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[176] D. R. Radev and K. McKeown. Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics*, 24(3):469–500, 1998.

[177] A. Ratnaparkhi. A Maximum Entropy Part–of–speech Tagger. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*, 1996.

[178] A. Ratnaparkhi. A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*, Brown University, Providence, RI, 1997.

[179] A Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Phd. Thesis, University of Pennsylvania, 1998. http://www.cis.upenn.edu/~adwait.

[180] A. Ratnaparkhi, J. Reynar, and S. Roukos. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 250–255, 1994.

[181] C. Reynar and A. Ratnaparkhi. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, Washington DC, 1997. ACL.

[182] German Rigau, Jordi Atserias, and Eneko Agirre. Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Joint ACL/EACL*, pages 48–55, Madrid, Spain, July 1997.

[183] E. Riloff and W. Lehnert. Information Extraction as a Basis for High–precision Text Classification. *ACM Transactions on Information Systems*, 12(3):296–333, 1994.

[184] R. L. Rivest. Learning Decision Lists. *Machine Learning*, 2:229–246, 1987.

[185] J. Rocchio. *Relevance Feedback in Information Retrieval.* G. Salton (editor), The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice–Hall, Inc., Englewood Cliffs, NJ, 1971.

[186] E. Roche and Y. Schabes. Deterministic Part–of–speech Tagging with Finite State Transducers. *Computational Linguistics*, 21(2):227–253, 1995.

[187] R. Rosenfeld. *Adaptive Statistical Language Modelling: A Maximum Entropy Approach.* Phd. Thesis, School of Computer Science, Carnegie Mellon University, 1994.

[188] D. Roth. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proceedings of the National Conference on Artificial Intelligence, AAAI '98*, July 1998. http://l2r.cs.uiuc.edu/~danr.

[189] D. Roth and D. Zelenko. Part of Speech Tagging Using a Network of Linear Separators. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 1136–1142, Montréal, Canada, 1998. http://l2r.cs.uiuc.edu/~danr.

[190] M. Sahami. Learning Limited Dependence Bayesian Classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338. AAAI Press, 1996.

[191] Ken Samuel. Lazy Transformation-Based Learning. In *Proceedings of the 11th International Florida Artificial Intelligence Research Symposium Conference*, pages 235–239, 1998. cmp–lg/9806003.

[192] Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. An Investigation of Transformation-Based Learning in Discourse. In *Proceedings of the 15th International Conference on Machine Learning, ICML'98*, 1998. cmp–lg/9806006.

[193] Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. Dialogue Act Tagging with Transformation-Based Learning. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 1150–1156, Montréal, Canada, 1998. cmp–lg/9806006.

[194] C. Samuelsson. Morphological Tagging Based Entirely on Bayesian Inference. In *Proceedings of the 9th Nordic Conference of Computational Linguistics*, Stockholm, Sweden, 1993.

[195] T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data*. Springer–Verlag, New York, 1989.

[196] R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[197] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, to appear. Also appearing in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 1998.

[198] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval, SIGIR '98*, 1998.

[199] H. Schmid. Part–of–speech Tagging with Neural Networks. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING*, pages 172–176, Kyoto, Japan, 1994.

[200] H. Schmid. Probabilistic Part–of–speech Tagging Using Decision Trees. In *Proceedings of the Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.

[201] H. Schütze. Part–of–speech Induction from Scratch. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 251–258, Columbus, OH, 1993. ACL.

[202] H. Schütze. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–124, 1998.

[203] H. Schütze and O. Pedersen. Information Retrieval based on Word Senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, NV, 1995.

[204] H. Schwenk and Y. Bengio. Training Methods for Adaptive Boosting of Neural Networks for Character Recognition. *Advances in Neural Information Processing Systems*, 10, 1998.

[205] F. Segond, A. Schiller, G. Grefenstette, and J-P. Chanod. An Experiment in Semantic Tagging using Hidden Markov Model Tagging. In *Proceedings of the Joint ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, pages 78–81, Madrid, Spain, 1997.

[206] T. J. Sejnowski and C. S. Rosenberg. Parallel Networks that Learn to Pronounce. *Complex Systems*, 1:145–168, 1987.

[207] S. Sekine. The Domain Dependence of Parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, Washington DC, 1997. ACL.

[208] H. Shinnou. Detection of japanese homophone errors by a decision list including a written word as a default evidence. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, Bergen, Norway, 1999.

[209] E. V. Siegel. Learning Methods for Combining Linguistic Indicators to Classify Verb. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP*, Brown University, Providence, RI, 1997. cmp-lg/9707015.

[210] R. F. Simmons and Y. Yu. The Acquisition and Use of Context–dependent Grammars for English. *Computational Linguistics*, 18(4):391–418, 1992.

[211] W. Skut and T. Brants. A Maximum–Entropy Partial Parser for Unrestricted Text. In *Proceedings of the 6th Workshop on Very Large Corpora*, Montréal, Canada, August 1998. cmp–lg/9807006.

[212] W. Skut and T. Brants. Chunk Tagger – Statistical Recognition of Noun Phrases. In *Proceedings of the ESSLLI'98 Workshop on automated Acquisition of Syntax and Parsing*, University of Saarbrücken, 1998. cmp–lg/9807007.

[213] S. Slattery and M. Craven. Combining Statistical and Relational methods for Learning in Hypertext Domains. In D. Page, editor, *Proceedings of the 8th International Workshop on Inductive Logic Programming*, pages 38–52. Springer, Berlin, 1998.

[214] T. C. Smith and I. H. Witten. Learning Language Using Genetic Algorithms. In *Proceedings of the IJCAI Workshop in New Approaches for NLP*, 1995. Also in S. Wermter, E. Riloff and G. Scheler (editors), Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing. Computer Notes in Artificial Intelligence 1040, Springer 1996.

[215] S. Soderland. Learning Information Extraction Rules for Semi–structured and Free Text. *Machine Learning*, 34:233–272, 1999.

[216] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. Crystal: Inducing a Conceptual Dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1314–1319, 1995.

[217] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. Issues in Inductive Learning of Domain–specific Text Extraction Rules. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040, pages 290–301. Springer, 1996.

[218] J. M. Sopena, A. Lloberas, and J. López. A Connectionist Approach to Prepositional Phrase Attachment. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 1233–1237, Montréal, Canada, August 1998.

[219] M. Srinivas and L. M. Patnaik. Genetic Algorithms: A survey. *Computer*, 27(6):17–26, 1994.

[220] C. Stanfill and D. Waltz. Toward memory–based reasoning. *Comunications of the ACM*, 29(12):1213–1228, 1986.

[221] H. Tanaka. Decision Tree Learning Algorithm with Structured Attributes: Application to Verbal Case Frame Acquisition. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING*, pages 943–948, Copenhagen, Denmark, August 1996.

[222] C. A. Thompson, R. J. Mooney, and L. R. Tang. Learning to Parse Natural Language Database Queries into Logical Form. In *Proceedings of the ML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition*, 1997.

[223] E. Tjong. Noun Phrase Representation by System Combination. In *Proceedings of the joint ANLP–NAACL 2000*, Seattle, Washington, USA, 2000. Morgan Kaufmann.

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

[225] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer–Verlag, New York, 1995.

[226] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, inc., New york, 1998.

[227] J. Veenstra. Fast NP Chunking Using Memory-based Learning Techniques. In *Proceedings of Benelearn*, Wageningen, the Netherlands, 1998. To appear.

[228] J. Véronis. A study of polysemy judgements and inter–annotator agreement. In *Programme and advanced papers of the Senseval workshop*, Herstmonceux Castle, England, 1998.

[229] V. Weber and S. Wermter. Using Hybrid Connectionist Learning for Speech/Language Analysis. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[230] Sholom M. Weiss, Chidanand Apte, Fred J. Damerau, David E. Johnson, Frank J. Oles, Thilo Goetz, and Thomas Hampp. Maximizing text-mining performance. *IEEE Intelligent Systems*, 14(4):63–69, 1999.

[231] S. Wermter, E. Riloff, and G. Scheler (editors). *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing.* Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[232] B. Widrow and S. D. Stearns. *Adaptive Signal Processing.* Prentice–Hall, Englewood Cliffs, NJ, 1985.

[233] E. J. Wiener, J. Pedersen, and A. Weigend. A Neural Network Approach to Topic Spotting. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995.

[234] K. Wnek, E. Bloedorn, and R. Michalski. Selective Inductive Learning Method AQ15C: The Method and User's Guide. Laboratory Report ML95-4, Machine Learning and Inference Laboratory, George Mason University, Fairfax, Virginia, 1995.

[235] T. Yamazaki, M. J. Pazzani, and C. Merz. Acquiring and Updating Hierarchical Knowledge for Machine Translation based on a Clustering Technique. In E. Riloff S. Wermter and G. Scheler, editors, *Connectionist, Statistical an Symbolic Approaches to Learning for Natural Language Processing*, Lecture Notes in Artificial Intelligence, 1040. Springer, 1996.

[236] J. J. Yang. *Use of Genetic Algorithms for Query Improvement in Information Retrieval Based on a Vector Space Model.* Phd. Thesis, University of Pittsburgh, Pittsburgh, PA, 1993.

[237] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1/2(1):67–88, 1999.

[238] Y. Yang and C. G. Chute. An Example–based Mapping Method for Text Classification and Retrieval. *ACM Transactions on Information Systems*, 12(3), 1994.

[239] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

[240] D. Yarowsky. One Sense per Collocation. In *DARPA Workshop on Human Language Technology*, Princeton, 1993.

[241] 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*, pages 88–95, Las Cruces, NM, 1994. ACL.

[242] 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*, pages 88–95, Las Cruces, NM, 1994. ACL.

[243] D. Yarowsky. Homograph Disambiguation in Speech Synthesis. In *Proceedings of 2nd ESCA/IEEE Workshop on Speech Synthesis*, New Paltz, NY, 1994.

[244] S. Young and G. Bloothooft (editors). *Corpus–based Methods in Language and Speech Processing.* An ELSNET book. Kluwer Academic Publishers, Dordrecht, 1997.

[245] J. Zavrel and W. Daelemans. Memory–Based Learning: Using Similarity for Smoothing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. Joint ACL/EACL*, Madrid, Spain, July 1997.

[246] J. Zavrel, W. Daelemans, and J. Veenstra. Resolving PP attachment Ambiguities with Memory–Based Learning. In *Proceedings of the Conference on Computational Natural Language Learning, CoNLL97*, pages 136–144, Madrid, Spain, 1997.

[247] J. M. Zelle and R. J. Mooney. Learning Semantic Grammars with Constructive Inductive Logic Programming. In *Proceedings of the 11th National Conference on Artificial Intelligence, AAAI*, pages 817–822, 1993. AAAI Press / MIT Press.

[248] J. M. Zelle and R. J. Mooney. Inducing Deterministic Prolog Parsers from Treebanks. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI*, pages 748–753, 1994. AAAI Press / MIT Press.

[249] J. M. Zelle and R. J. Mooney. Learning to Parse Database Queries Using Inductive Logic. In *Proceedings of the 13th National Conference on Artificial Intelligence, AAAI '96*, pages 1050–1055, Portland, OR, 1996.

A Technical Details

A.1 SNoW in Detail

SNoW [89] stands for Sparse Network of Winnows. It is a on-line learning system which has the Winnow algorithm as the basic component. This method has not been applied to WSD before. For this reason it will be described in more detail.

Winnow Algorithm

Winnow [123] is a linear threshold algorithm for 2-class problems with binary (i.e., 0/1-valued) input features. It classifies a new example x into positive class if

$$\sum w_f > \theta, \ \forall f \in F$$

and into negative class otherwise. In this formulation, F is the set of *active features* of x (i.e. those that have value 1), w_f is the weight associated to input feature f and θ is the *threshold parameter*. Winnow is an online algorithm; it accepts examples one-at-a-time and updates the weights w_j as necessary.

Winnow initializes its weights w_f to 1. It then accepts a new training example (x,y), where y is the class of x, and applies the threshold rule to compute the predicted class y'. Winnow is a mistake–driven algorithm, that is, it updates its hypotheses only when a mistake is made. If the predicted class is correct ($y = y'$), Winnow does nothing. However, if the predicted class is wrong, Winnow updates its weights as follows. If y' is negative and y positive, then the weights of active features are promoted (informally, they were too low):

$$\forall f \in F, \ w_f \leftarrow \alpha \cdot w_f$$

where α is a number greater than 1 called *promotion parameter*. If y' is positive and y negative, then the weights corresponding active features are demoted (informally, they were too high):

$$\forall f \in F, \ w_f \leftarrow \beta \cdot w_f$$

where β is a positive number lower than 1 called *demotion parameter*.

Snow Architecture

In the SNoW architecture there is a Winnow node for each class, which learns to separate that class from all the rest. Figure 4 shows the SNoW architecture learned to disambiguate between the two senses of *age* presented in table 6.

Initially, there are no connections between features and classes. During the training step, examples are processed sequentially. Each training example is a positive example for the class it belongs and negative for the rest. A key point that allows a fast learning is that the Winnow nodes are not connected to all features but only to those that are "relevant" for their class (i.e. those that appear in at least one training example of that class). For instance, in figure 4, the features "$w_{+1} = average$" and "$w_{+2} = 42$" are connected only to the first node (and "$w_{+2} = nuclear$" only to the second node), while feature "$w_{+1} = of$" is connected to both winnow nodes because it appears in examples of both classes. These connections between features and classes are created dynamically as the learning examples are processed.

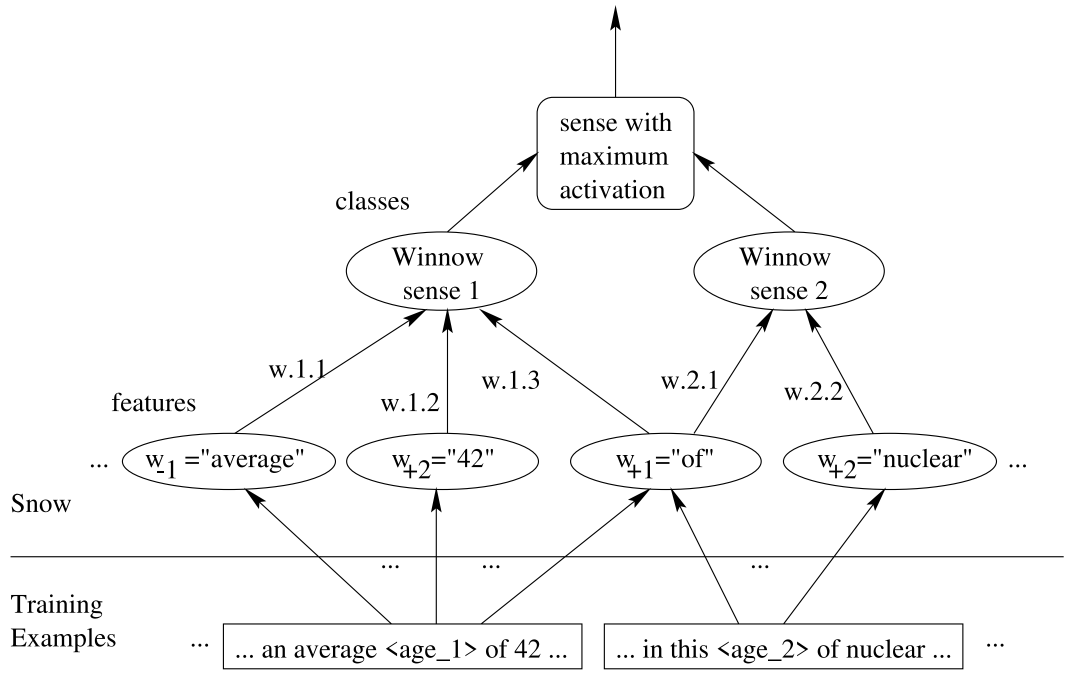

Figure 4: SNoW Architecture

In [62] the training step is repeated several times so as to better adjust the weights of the network. In order to clarify this, these training step repetitions are equivalent to the Neural Network learning epochs. However, the utility of that retraining step is dubious, given that it is not even mentioned in some papers in which SNoW is applied [189, 89]. Retraining implies to add a new parameter to the system, "when to stop". In [62] the training process finishes when:

1. no mistakes are made on the training examples; or

2. a maximum number of epochs is achieved (say 50).

When classifying a new example, SNoW is similar to a neural network which takes the input features and outputs the class with the highest activation.

Table 12 shows the particular parameter values used in the following experiments[24]. These values have been empirically determined and they are those that lead to better accuracy of the algorithm in the WSD task.

SNoW is proven to perform very well in high dimensional domains, where both, the training examples and the target function reside very sparsely in the feature space [188], such as text categorization [62] or context–sensitive spelling correction [89].

A.2 AdaBoost.MH **in Detail**

This section describes the Schapire and Singer's AdaBoost.MH algorithm for multiclass multi–label classification, using exactly the same notation given by the authors in [198, 196].

[24] When a new connection between a feature and a class is activated its weight is initialized to the value $\frac{\theta}{\|F\|}$, which is the threshold value divided by the number of active features of the current example.

Parameter	Value
Promotion (α)	1.5
Demotion (β)	0.8
Threshold (θ)	1
Initial weights	$\frac{\theta}{\lVert F \rVert}$
Number of training epochs	1

Table 12: SNoW Parameter Values

As already said, the purpose of boosting is to find a highly accurate classification rule by combining many **weak hypotheses** (or weak rules), each of which may be only moderately accurate. It is assumed that there exists a separate procedure called the WeakLearner for acquiring the weak hypotheses. The boosting algorithm finds a set of weak hypotheses by calling the weak learner repeatedly in a series of T rounds. These weak hypotheses are then combined into a single rule called the **combined hypothesis.**

Let $S = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_m, Y_m)\}$ be the set of m training examples, where each instance x_i belongs to an instance space $\mathcal{X}$ and each Y_i is a subset of a finite set of labels or classes $\mathcal{Y}$. The size of $\mathcal{Y}$ is denoted by $k = |\mathcal{Y}|$.

The pseudo–code of AdaBoost.MH is presented in figure 5.[25]

AdaBoost.MH maintains an $m{\times}k$ matrix of weights as a distribution D over examples and labels. The goal of the WeakLearner algorithm is to find a weak hypothesis with moderately low error with respect to these weights. Initially, the distribution D_1 is uniform, but the boosting algorithm updates the weights on each round to force the weak learner to concentrate on the pairs (examples,label) which are hardest to predict.

procedure AdaBoost.MH (**in**: $S = \{(\mathbf{x}_i, Y_i)\}_{i=1}^{m}$)
S is the set of training examples
Initialize distribution D_1 (for all i, $1 \le i \le m$, and all l, $1 \le l \le k$)
$D_1(i,l) = 1/(mk)$
for t:=1 **to** T **do**
Get the weak hypothesis $h_t : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$
h_t = WeakLearner (X, D_t);
Update distribution D_t (for all i, $1 \le i \le m$, and all l, $1 \le l \le k$)

$$D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-Y_i[l]h_t(x_i,l))}{Z_t}$$

Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution)
end-for
return the combined hypothesis: $f(\mathbf{x}, l) = \sum_{t=1}^{T} h_t(\mathbf{x}, l)$
end AdaBoost.MH

Figure 5: The AdaBoost.MH algorithm

[25] The general formulation of AdaBoost.MH takes into account an additional parameter α_t. However, the way in which the WeakLearner will be defined implies that α_t must be set to 1 [196], and so it has no effect.

More precisely, let D_t be the distribution at round t, and $h_t : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ the weak rule acquired according to D_t. The sign of $h_t(\mathbf{x}, l)$ is interpreted as a prediction of whether label l should be assigned to example $\mathbf{x}$ or not. The magnitude of the prediction $|h_t(\mathbf{x}, l)|$ is interpreted as a measure of confidence in the prediction. In order to understand correctly the updating formula this last piece of notation should be defined. Thus, given $Y \subseteq \mathcal{Y}$ and $l \in \mathcal{Y}$, let $Y[l]$ be +1 if $l \in Y$ and -1 otherwise.

Now, it becomes clear that the updating function increases (or decreases) the weights $D_t(i, l)$ for which h_t makes a good (or bad) prediction, and that this variation is proportional to $|h_t(\mathbf{x}, l)|$.

Note that WSD is not a multi–label classification problem since a unique sense is expected for each word in context. In our implementation, the algorithm runs exactly in the same way as explained above, except that sets Y_i are reduced to a unique label, and that the combined hypothesis is forced to output a unique label, which is the one that maximizes $f(x, l)$.

Up to now, it only remains to be defined the form of the WeakLearner. Schapire and Singer [198] prove that the Hamming loss of the AdaBoost.MH algorithm on the training set[26] is at most $\prod_{t=1}^{T} Z_t$, where Z_t is the normalization factor computed on round t. This upper bound is used in guiding the design of the WeakLearner algorithm, which attempts to find a weak hypothesis h_t that minimizes:

$$Z_t = \sum_{i=1}^{m} \sum_{l \in \mathcal{Y}} D_t(i, l) \exp(-Y_i[l] h_t(\mathbf{x}, l)) \, .$$

Weak Hypotheses for WSD

As in [2], very simple weak hypotheses are used to test the value of a boolean predicate and make a prediction based on that value. The predicates used, which are described in section 3.5, are of the form "$f = v$", where f is a feature and v is a value (e.g.: "previous_word = *hospital*"). Formally, based on a given predicate p, our interest lies on weak hypotheses h which make predictions of the form:

$$h(\mathbf{x}, l) = \begin{cases} c_{0l} & \text{if } p \text{ holds in } \mathbf{x} \\ c_{1l} & \text{otherwise} \end{cases}$$

where the c_{jl}'s are real numbers.

For a given predicate p, and bearing the minimization of Z_t in mind, values c_{jl} should be calculated as follows. Let X_1 be the subset of examples for which the predicate p holds and let X_0 be the subset of examples for which the predicate p does not hold. Let $[\![\pi]\!]$, for any predicate π, be 1 if π holds and 0 otherwise. Given the current distribution D_t, the following real numbers are calculated for each possible label l, for $j \in \{0, 1\}$, and for $b \in \{+1, -1\}$:

$$W_b^{jl} = \sum_{i=1}^{m} D_t(i, l) [\![\mathbf{x}_i \in X_j \wedge Y_i[l] = b]\!]$$

That is, W_{+1}^{jl} (W_{-1}^{jl}) is the weight (with respect to distribution D_t) of the training examples in partition X_j which are (or not) labelled by l.

[26] i.e. the fraction of training examples i and labels l for which the sign of $f(\mathbf{x}_i, l)$ differs from $Y_i[l]$.

As it is shown in [198], Z_t is minimized for a particular predicate by choosing:

$$c_{jl} = \frac{1}{2}\ln(\frac{W_{+1}^{jl}}{W_{-1}^{jl}})$$

These settings imply that:

$$Z_t = 2 \sum_{j \in \{0,1\}} \sum_{l \in \mathcal{Y}} \sqrt{W_{+1}^{jl} W_{-1}^{jl}}$$

Thus, the predicate p chosen is that for which the value of Z_t is smallest.

Very small or zero values for the parameters W_b^{jl} cause c_{jl} predictions to be large or infinite in magnitude. In practice, such large predictions may cause numerical problems to the algorithm, and seem to increase the tendency to overfit. As suggested in [196], smoothed values for c_{jl} have been used.