In Proceedings of the 2nd Indian International conference on Artificial Intelligence (IICAI-05), December 20-22 2005, Pune, India.

A Comparative Study of Support Vector Machines Applied to the Supervised Word Sense Disambiguation Problem in the Medical Domain

Mahesh Joshi, Ted Pedersen and Richard Maclin
{joshi031, tpederse, rmaclin}@d.umn.edu

Department of Computer Science
University of Minnesota, Duluth, MN 55812, USA

Abstract. We have applied five supervised learning approaches to word sense disambiguation in the medical domain. Our objective is to evaluate Support Vector Machines (SVMs) in comparison with other well known supervised learning algorithms including the naïve Bayes classifier, C4.5 decision trees, decision lists and boosting approaches. Based on these results we introduce further refinements of these approaches. We have made use of unigram and bigram features selected using different frequency cut-off values and window sizes along with the statistical significance test of the log likelihood measure for bigrams. Our results show that overall, the best SVM model was most accurate in 27 of 60 cases, compared to 22, 14, 10 and 14 for the naïve Bayes, C4.5 decision trees, decision list and boosting methods respectively.

1 Introduction

English has many words that have multiple meanings or multiple senses. For example, the word *switch* in the sentence *Turn off the main switch* refers to an electrical instrument whereas in the sentence *The hansom driver whipped the horse using a switch* it refers to a flexible twig or rod[1]. As can be observed in these examples, the correct sense of the word *switch* is made clear by the context in which the word has been used. Specifically, in the first sentence, the words *turn, off* and *main* combined with some world knowledge of the person interpreting the sentence such as the fact that usually there is a main switch for electrical connections inside a house, help in disambiguating the word (i.e., assigning the correct sense to the word). Similarly, in the second sentence the words *hansom, driver, whipped* and *horse* provide the appropriate context which helps in understanding the correct sense of the word *switch* for that sentence.

Word sense disambiguation (WSD) is the problem of automatically assigning the appropriate meaning to a word having multiple senses. As noted earlier, this process relies to a great extent on the surrounding context of the word and analyzing the properties exhibited by that context.

[1] According to the Merriam-Webster Online Dictionary: http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=switch

It is sometimes theorized that ambiguity is less of a problem in more specialized domains. However, we have observed that ambiguity remains a problem even in the specialized domain of medicine. For example, *radiation* could be used to mean the property of *electromagnetic radiation*, or as a synonym for *Radiation therapy* for treatment of a disease. While both of these senses are somewhat related (the therapy relies on the radioactive property) there are also cases such as *cold*, which can mean the temperature of a room, or an illness. Thus, even more specialized domains exhibit a full range of ambiguities.

As noted by Weeber et al. [15], linguistic interest in medical domain arises out of the need for better natural language processing (NLP) systems used for decision support or document indexing for information retrieval. Such NLP systems will perform better if they are capable of resolving ambiguities among terms. For example, with the ability to disambiguate senses, an information retrieval query for *radiation therapy* would focus on those documents that contain the word *radiation* in the "medical treatment" sense.

Most work in word sense disambiguation has focused only on general English. Here we propose to study word sense disambiguation in the medical domain and evaluate how well existing techniques perform, and introduce refinements of our own based on this experience. The intuition behind experimenting with existing approaches is the following – although the ambiguity in the medical domain might tend to focus around domain specific terminology, the basic problems it poses for sense distinction may not be strikingly different from those encountered in general English sense distinction.

The most popular approaches in word sense disambiguation have been those that rely on supervised learning. These methods initially train a machine learning algorithm using various instances of the word which are manually tagged with the appropriate sense. The result of this training is a classifier that can be applied to future instances of the ambiguous word. Support Vector Machines [14] are one such class of machine learning algorithms. While SVMs have become popular for use in general English word sense disambiguation, they have not been explored in the domain of medical text. Our objective is to see if the good performance of SVMs in general English will translate into this new domain and also to compare SVM performance with some other well known machine learning algorithms.

This paper continues with a description of related work in Section 2 and a brief background on machine learning methods in Section 3. Section 4 outlines our experimental setup and feature selection while Section 5 explains our evaluation methodology. Section 6 focuses on discussion of our results. We discuss the future work for this ongoing research in Section 7. Section 8 summarizes our work so far.

2 Related Work

In the last several years, a number of researchers have explored the use of Support Vector Machines in general English word sense disambiguation.

Cabezas et al. [3] present a supervised word sense tagger using Support Vector Machines. Their system was designed for performing word sense disambiguation independent of the language of lexical samples provided for the SENSEVAL-2 task. A lexical sample for an ambiguous word is a corpus containing several instances of that word, having multiple senses. Their system identified two types of features – (a) unigrams in a wider context of the ambiguous word and (b) up to three words on either side of the ambiguous word with their orientation and distance with respect to the ambiguous word. The second feature captures the *collocations* containing the ambiguous word, in a narrow context around the word. Cabezas et al. use the term collocations to mean *word co-occurrences* unlike the more conventional linguistic sense which defines collocations as *two or more words that occur together more often than by chance*. These features were weighed according to their relevance for each ambiguous word, using the concept of Inverse Category Frequency (ICF) where the ICF score of a feature is higher when it is more representative of any particular sense. For multi-class classification of words having more than two senses, they employed the technique of building a "one against all" classifier for each of the senses. In this method, the classifier for a given sense categorizes all the instances into two classes – one that represents the given sense and the other that represents anything that does not belong to the given sense. For any ambiguous word, the sense that is assigned is the one whose classifier voted for that sense with highest confidence. Their results show a convincing improvement over baseline performance.

Lee et al. [8] use Support Vector Machines to perform Word Sense Disambiguation for general English and for translating an ambiguous English word into its Hindi equivalent. They have made use of all the features available from the following knowledge sources: (a) Parts Of Speech (POS) of up to three words around the ambiguous word and POS of the ambiguous word itself, (b) morphological root forms of unigrams in the entire context, with function words, numbers and punctuations removed, (c) collocations, that is word co-occurrences consisting of up to three words around the ambiguity and (d) various syntactic relations depending upon the POS of the ambiguous word. They make use of all the extracted features and do not perform any kind of feature selection, that is they do not use any statistical or information gain measures to refine their feature set. Additionally, they have also used (e) the English sense of ambiguous words as a feature for the translation task, which improved their system's performance. They have made use of the SVM implementation available in the Weka data mining suite [16], with the linear kernel and default parameter values. This is the exact configuration that we have used for our experiments. The results that they obtained for the general English corpus were better than those obtained for the translation task.

Ngai et al. [11] propose a supervised approach to semantic role labeling. The FrameNet corpus [1] is an ontologically structured lexical database that consists of semantic frames, lexical units that activate these frames, and a large corpus of annotated sentences belonging to the various semantic frames. A *semantic frame* is an abstract structure relating to some event or concept and includes

the participant objects of the event or concept. These participant objects are known as *frame elements*. Frame elements are assigned semantic types wherever appropriate. A *lexical unit* is any word in a sentence (often the verb, but not necessarily so) that determines the semantic frame the sentence belongs to. For example, FrameNet consists of a semantic frame titled *Education_teaching*, two of its frame elements being *Teacher* and *Student* which have the semantic type *Sentient*. Some of the lexical units which activate this frame are *coach, educate, education, teach* and *instruct*. Ngai et al. propose to solve the problem of semantic role labeling of sentence parse constituents by posing it as a classification task of assigning the parse constituents to the appropriate frame element from the FrameNet corpus. This is in principle similar to our task where we aim at classifying words into different concepts as defined in the Unified Medical Language System (UMLS) repository, which is to some extent more "coarse" than word sense disambiguation in the conventional sense. They make use of the following types of features: *(a)* lexical and syntactic features available from the FrameNet ontology – such as the lexical identity of the target word, its POS tag, syntactic category and *(b)* extracted features such as the transitivity and voice of verbs, and head word of the parse constituent. They have tested different machine learning methods including boosting, SVMs, maximum entropy, Sparse Network of Winnows (SNOW) and decision lists – individually as well as their ensembles (i.e., additive learning methods). Their best results from SVMs were obtained with polynomial kernel with degree four. For multi-class classification, they too have used the "one against all" approach. Although SVMs were not the best individually due to their comparatively lower recall scores, they obtained very high precision values and were part of the classifier ensemble that gave the best results.

Recently Gliozzo et al. [6] have presented domain kernels for word sense disambiguation. The key notion is to make use of domain knowledge while performing word sense disambiguation. An example they discuss is the ambiguity of the word *virus*. A *virus* can mean *"a malicious computer program"* in the domain of computers or *"an infectious agent which spreads diseases"* if we switch to the domain of medicine. Gliozzo et al. propose a domain matrix (with words along the rows and domains along the columns) that consists of soft clusters of words in different domains. A word can belong to multiple domains with different probabilities – thus representing word ambiguity, whereas a domain can contain multiple words – thus representing its variability. They make use of the fully unsupervised approach of Latent Semantic Analysis (LSA) to automatically induce a domain matrix from raw text corpus. This domain matrix is used in transforming the conventional *term by document* vector space model into a *term by domain* vector space model, where the domains are the ones induced by LSA. This is called the domain vector space model. They define a domain kernel function which evaluates distances among two words by operating upon the corresponding word vectors obtained from this domain vector space model. Traditionally these vectors are created using Bag Of Words (BOW) or POS features of words in surrounding context. The kernels using these traditional

vectors are referred as the BOW kernel and the POS kernel respectively. Using the domain kernels, Gliozzo et al. have demonstrated significant improvement over BOW and POS kernels. By augmenting the traditional approaches with domain kernels, their results show that only 50 percent of the training data is required in order to attain the accuracy offered by purely traditional approaches, thus reducing the knowledge acquisition bottleneck to a great extent.

The National Library of Medicine (NLM) WSD collection is a set of 50 ambiguous medical terms collected from medical journal abstracts. It is a fairly new dataset and has not been explored much. Following is related work which makes use of this collection.

Liu et al. [10] evaluate the performance of various classifiers on two medical domain datasets and one general English dataset. The classifiers that they have considered included the traditional decision lists, their adaptation of the decision lists, the naïve Bayes classifier and a mixed learning approach that they have developed. Their features included combinations of (a) unigrams in various window sizes around the ambiguous word with their orientation and distance information and (b) two-word collocations (word co-occurrences) in a window size of two on either side of the ambiguous word, and not including the ambiguous word. The general biomedical term dataset that they used is a sub-set of the NLM WSD data collection that we have used for our experiments. They achieved best results for the medical abbreviation dataset using their mixed learning approach and the naïve Bayes classifier. No particular combination of features, window size and classifiers provided stable performance for all the ambiguous terms. They therefore concluded that the various approaches and feature representations were complimentary in nature and as a result their mixed learning approach was relatively stable and obtained better results in most of the cases.

Leroy and Rindflesch [9] explore the use of symbolic knowledge from the UMLS ontology for disambiguation of a subset of the NLM WSD collection. The basic features of the ambiguous word that they use are (a) status of the ambiguous word in the phrase – whether it is the main word or not, and (b) its part of speech. Unlike many BOW approaches which use the actual words in context as features, they make use of (c) semantic types of words in the context as features. Additionally they use (d) semantic relations among the semantic types of non-ambiguous words. Finally, they also make use of the (e) semantic relations of the ambiguous type with its surrounding types. The semantic types and their relations are derived from the UMLS ontology. Using the naïve Bayes classifier from the Weka data mining suite [16], their experiments were performed with incremental feature sets, thus evaluating the contribution of new features over the previous ones. They achieved convincing improvements over the majority sense baseline in some cases, but observed degradation of performance in others. In general it was not the case that a maximum set of featurs yielded the best results. However, semantic types of words in context and their relationship with the various senses of the ambiguous word were useful features along with the information whether the ambiguous word was the main word or not. Therefore

this approach can possibly be used in combination with the conventional BOW approaches to improve the results.

3 Machine Learning Methods

Support Vector Machines (SVM) [14] represent data instances in an N dimensional *hyperspace* where N represents the number of features identified for each instance. The goal of an SVM learner is to find a *hyperplane* that separates the instances into two distinct classes, with the maximum possible separation between the hyperplane and the nearest instance on both sides. The maximum separation helps to achieve better generalization on unknown input data. The nearest correctly classified data point(s) on either side of the hyperplane are known as *support vectors* to indicate that they are the crucial points which determine the position of the hyperplane. In the event that a clear distinction between data points is not possible, a penalty measure known as a slack variable is introduced to account for each instance that is classified incorrectly. Mathematically, SVM classification poses an optimization problem in which an equation is to be minimized, subject to a set of linear constraints. Due to this, the training time for SVMs is often high. As a result, various approaches have been developed to enhance the performance of SVMs. One such algorithm that effectively works around the time consuming step of numerical quadratic programming is the Sequential Minimal Optimization (SMO) [12] algorithm. We use the Weka [16] implementation of the SMO algorithm for our experiments. This implementation uses the "pairwise coupling" [7] technique for multi-class classification problems. In this method, one classifier is created for each pair of the target classes, ignoring instances that belong to other classes. For example, with three classes C_1, C_2, and C_3, three classifiers for the pairs $\{C_1, C_2\}$, $\{C_2, C_3\}$ and $\{C_3, C_1\}$ are trained using data instances that belong to the two respective classes. The output of each pairwise classifier is a probability estimate for its two target classes. The pairwise probability estimates from all the classifiers are combined together to come up with an absolute probability estimate for each class.

The naïve Bayes classifier is based on a probabilistic model of conditional independence. It calculates the posterior probability that an instance belongs to a particular class given the prior probabilities of the class and the feature set that is identified for each of the instances. The "naïve" part of the classifier is that it assumes that each of the features for an instance are conditionally independent – meaning that given a particular class, the presence of one feature does not affect the likelihood of occurrence of other features for that class. Given the features F_1 and F_2, the equality in Equation 1 gives the posterior probability of class C_i according to the Bayes rule. The naïve Bayes classifier makes the subsequent approximation of assuming that the features are conditionally independent. After calculating the posterior probabilities for each of the classes, it assigns the instance to the class with the highest posterior probability.

$$P(C_i|F_1, F_2) = \arg\max_i \frac{P(F_1, F_2|C_i)}{P(F_1, F_2)} \approx \arg\max_i P(F_1|C_i).P(F_2|C_i) \quad (1)$$

The C4.5 decision tree [16] learning approach is based on the "Divide and Conquer" strategy. The classifier constructs a decision tree where each node is a test of some feature, progressing from the top to the bottom, that is from the root to the leaves. Therefore, the higher the node is in the hierarchy the more crucial is the feature that is evaluated at that node while deciding the target class. The nodes of the tree are selected in such a way that the one which presents the maximum gain of information for classification is higher in the hierarchy. Additionally, the C4.5 algorithm includes handling of numerical attributes, missing values and pruning techniques to reduce the size and complexity of a decision tree.

Decision list learning is a rule-based approach, essentially consisting of a set of conditional statements like "if-then" or "switch-case" conditions for classifying data. These rules are applied in sequence until a condition is found to be true and the corresponding class is returned as the output. In case of failure of all rules, these classifiers return the class with the most frequent occurrence, in the case of WSD – the majority sense. Frank and Witten [4] discuss an approach of repeatedly building partial decision trees to generate a decision list. Their algorithm avoids the conventional two-step procedure of initially building a list of rules and then processing them in a second step for pruning and optimization.

The Boosting approach to machine learning [13] is to combine a set of weaker classifiers obtained by repeatedly running an elementary base classifier on different sub-sets of training data. The idea is that obtaining elementary classifiers that give reasonable performance is simpler than trying to find one complex classifier that fits all of the data points. Combining these weak classifiers into one single prediction strategy often achieves significantly better performance than any of the weak classifiers can individually achieve. We use the Weka implementation of the AdaBoost.M1 algorithm, which is a multi-class extension of the AdaBoost algorithm proposed by Freund and Schapire [5]. The base classifier in our experiments is the DecisionStump classifier, which is a single node decision tree classifier that tests just one feature and predicts the output class.

We use off-the-shelf implementations of all of the above algorithms, which are available in the Weka data mining suite [16]. We retain the default settings for all the classifiers and carry out ten-fold cross-validation.

4 Experimental Setup

4.1 Data

We have made use of the biomedical word sense disambiguation test collection developed by Weeber et al. [15]. This WSD test collection is available from the National Library of Medicine (NLM).[2] The Unified Medical Language System (UMLS) [3] consists of three knowledge sources related to biomedicine and health: (1) the metathesaurus of biomedical and health related concepts such

[2] http://wsd.nlm.nih.gov/
[3] http://www.nlm.nih.gov/research/umls/about_umls.html

1|9337195.ab.7|M2
The relation between birth weight and flow-mediated dilation was not affected by ***adjustment*** for childhood body build, parity, cardiovascular risk factors, social class, or ethnicity.
adjustment|adjustment|78|90|81|90|by adjustment|
PMID- 9337195
TI - Flow-mediated dilation in 9- to 11-year-old children: the influence of intrauterine and childhood factors.
AB - BACKGROUND: Early life factors, particularly size at birth, may influence later risk of cardiovascular disease, but a mechanism for this influence has not been established. We have examined the relation between birth weight and endothelial function (a key event in atherosclerosis) in a population-based study of children, taking into account classic cardiovascular risk factors in childhood. METHODS AND RESULTS: We studied 333 British children aged 9 to 11 years in whom information on birth weight, maternal factors, and risk factors (including blood pressure, lipid fractions, preload and postload glucose levels, smoking exposure, and socioeconomic status) was available. A noninvasive ultrasound technique was used to assess the ability of the brachial artery to dilate in response to increased blood flow (induced by forearm cuff occlusion and release), an endothelium-dependent response. Birth weight showed a significant, graded,positive association with flow-mediated dilation (0.027 mm/kg; 95% CI, 0.003 to 0.051 mm/kg; P=.02). Childhood cardiovascular risk factors (blood pressure, total and LDL cholesterol, and salivary cotinine level) showed no relation with flow-mediated dilation, but HDL cholesterol level was inversely related (-0.067 mm/mmol; 95% CI, -0.021 to -0.113 mm/mmol; P=.005). The relation between birth weight and flow-mediated dilation was not affected by ***adjustment*** for childhood body build, parity, cardiovascular risk factors, social class, or ethnicity. CONCLUSIONS: Low birth weight is associated with impaired endothelial function in childhood, a key early event in atherogenesis. Growth in utero may be associated with long-term changes in vascular function that are manifest by the first decade of life and that may influence the long-term risk of cardiovascular disease.
adjustment|adjustment|1521|1533|1524|1533|by adjustment|

Fig. 1. A typical instance of an ambiguous term in the NLM WSD data collection. The example above shows an instance of the term *adjustment*.

as names of diseases or agents causing them, for example *Chronic Obstructive Airway Disease* and *Virus*. (2) The semantic network which provides a classification of these concepts and relationships among them. The relationships can be *hierarchical* as in "Acquired Abnormality **IsA** Anatomical Abnormality" or *associative* as in "Acquired Abnormality **affects** Cell Function". (3) The SPECIALIST lexicon containing biomedical terms with their syntactic, morphological, and orthographic information. MEDLINE (Medical Literature Analysis and Retrieval System Online) [4] is a bibliographic database containing references to several journals related to life science. The NLM WSD collection consists of 50 frequently encountered ambiguous words in the MEDLINE 1998 collection in the UMLS. While most of the words appear predominantly in noun form, there are also cases where they appear as adjectives or verbs. For example, the word *Japanese* occurs by itself as a noun meaning *the Japanese language* or *the Japanese people*, but more often as an adjective to describe people as in *the Japanese researchers* or *the Japanese patients*. Some words appear as verbs in their morphological variations, for example *discharge* appears as *discharged* and *determination* as *determined*. Each of the words has 100 randomly selected instances from the abstracts of 409,337 MEDLINE citations. Each instance provides two contexts for the ambiguous word – the sentence that contains the ambiguous word and the entire abstract that contains the sentence. The average size of the sentence context is 26 words and that of the abstract context is 220 words. The data is available in plain text format and follows some pre-defined formatting rules. Figure 1 shows a typical instance of an ambiguous term in the NLM WSD data collection. As noted earlier, one of the datasets used by Liu et al. [10] and the dataset used by Leroy and Rindflesch [9] were subsets of this collection.

[4] http://www.nlm.nih.gov/pubs/factsheets/medline.html

Table 1. Sense distribution for the ambiguous terms in the NLM WSD Collection, the sense frequencies are out of 100.

Word	**Senses**	**Sense tag frequency**					
		M1	M2	M3	M4	M5	None
adjustment	4	18	62	13	-	-	7
association	3	0	0	-	-	-	100
blood_pressure	4	54	2	44	-	-	0
cold	6	86	6	1	0	2	5
condition	3	90	2	-	-	-	8
culture	3	11	89	-	-	-	0
degree	3	63	2	-	-	-	35
depression	3	85	0	-	-	-	15
determination	3	0	79	-	-	-	21
discharge	3	1	74	-	-	-	25
energy	3	1	99	-	-	-	0
evaluation	3	50	50	-	-	-	0
extraction	3	82	5	-	-	-	13
failure	3	4	25	-	-	-	71
fat	3	2	71	-	-	-	27
fit	3	0	18	-	-	-	82
fluid	3	100	0	-	-	-	0
frequency	3	94	0	-	-	-	6
ganglion	3	7	93	-	-	-	0
glucose	3	91	9	-	-	-	0
growth	3	37	63	-	-	-	0
immunosuppression	3	59	41	-	-	-	0
implantation	3	17	81	-	-	-	2
inhibition	3	1	98	-	-	-	1
japanese	3	6	73	-	-	-	21

Tables 1 and 2 show the distribution of different senses for each word in the collection. *M1* through *M5* are different senses for a word as defined in the UMLS repository. Note that not every word has five senses defined in UMLS. Most of them have just two. The last column with the sense *None* stands for any sense other than *M1* thorough *M5*. The number of senses in the second column counts *None* as one of the senses. A few salient features that can be observed from the distribution are as follows. Every word has *None* as one of the possible senses – which means that while manually tagging the data instances, an instance which cannot be categorized into any of the known concepts as defined in UMLS can be assigned this default sense. Although the machine learning methods will see these instances as having the *same* sense, the features present in such instances will often be an entirely random mixture representing multiple other unknown senses. This effect will be more pronounced in the cases where the *None* sense covers almost 50 percent of the instances or greater. These instances introduce significant noise into the data. Therefore, for such words the performance of machine learning methods might degrade. Half of the words in the dataset have a majority sense that covers 80 percent of the instances, making their sense

Table 2. Sense distribution for the ambiguous terms in the NLM WSD Collection (continued from Table 1). The word *mosaic* has two senses that are very closely related and were assigned the same label *M2*.

Word	**Senses**	**Sense tag frequency**					
		M1	M2	M3	M4	M5	None
lead	3	27	2	-	-	-	71
man	4	58	1	33	-	-	8
mole	4	83	1	0	-	-	16
mosaic	4	45	**52**	*	0	-	3
nutrition	4	45	16	28	-	-	11
pathology	3	14	85	-	-	-	1
pressure	4	96	0	0	-	-	4
radiation	3	61	37	-	-	-	2
reduction	3	2	9	-	-	-	89
repair	3	52	16	-	-	-	32
resistance	3	3	0	-	-	-	97
scale	4	0	65	0	-	-	35
secretion	3	1	99	-	-	-	0
sensitivity	4	49	1	1	-	-	49
sex	4	15	5	80	-	-	0
single	3	1	99	-	-	-	0
strains	3	1	92	-	-	-	7
support	3	8	2	-	-	-	90
surgery	3	2	98	-	-	-	0
transient	3	99	1	-	-	-	0
transport	3	93	1	-	-	-	6
ultrasound	3	84	16	-	-	-	0
variation	3	20	80	-	-	-	0
weight	3	24	29	-	-	-	47
white	3	41	49	-	-	-	10

distribution highly skewed. Finally, a note regarding the word *mosaic*: two of its senses are very closely related – *M2* (Mosaicism) and *M3* (Embryonic Mosaic). They were therefore assigned the same label *M2* during manual sense tagging. This sense covers 52 instances, which are listed in the column *M2*.

4.2 Feature Selection

Before performing feature selection, we convert the NLM formatted data into SENSEVAL-2 format. SENSEVAL-2 format for WSD is an XML format with certain pre-defined markup tags. Figure 2 shows the partial contents of the generated SENSEVAL-2 files. For every word, two files are created, one containing the abstract context and other containing the sentence context for all of its instances. The feature selection programs that we use operate upon the lexical sample created out of combining the contexts (either abstract or sentence) for all of the instances of a given word. This lexical sample is then processed to remove punctuations and functional words or stop words. In addition to removing

common pre-defined functional words, we also remove any word that is entirely in upper case letters. This is done because many of the citations include headings like BACKGROUND, METHODS, RESULTS and CONCLUSIONS which introduce noise into the feature set by getting identified as significant features for many or all senses. Apart from this, we also eliminate any XML markup tags from the context data. Once this pre-processing is complete, we identify the following two types of features using the Ngram Statistics Package (NSP) [2].

(a) *Unigrams* : We identify the significant words occurring in the lexical sample for a word and use them as binary features. So a unigram feature vector for a given instance of an ambiguous word will consist of a sequence of ones and zeroes depending upon whether the corresponding unigram is present in the context of that instance or not. Currently, the only significance criteria that we apply for selecting the unigram features is a frequency cut-off value ranging from two to five. A frequency cut-off value of five means a unigram is discarded if it appears less than five times in the lexical sample.

(b) *Bigrams* : We select significant two-word collocations in the lexical sample and use them as binary features, similar to the unigrams. Bigrams can be

Abstract context in SENSEVAL2 format.

```
<corpus lang='en'>
 <lexelt item="adjustment">
  <instance id="9337195.ab.7" pmid="9337195" alias="adjustment">
   <answer instance="9337195.ab.7" senseid="M2"/>
   <context>
    <title>Flow-mediated dilation in 9- to 11-year-old children: the influence of intrauterine and childhood factors.
</title> BACKGROUND: Early life factors, particularly size at birth, may influence later risk of cardiovascular disease,
but a mechanism for this influence has not been established. We have examined the relation between birth weight
and endothelial function (a key event in atherosclerosis) in a population-based study of children, taking into account
classic cardiovascular risk factors in childhood. METHODS AND RESULTS: We studied 333 British children aged 9 to 11
years in whom information on birth weight, maternal factors, and risk factors (including blood pressure, lipid
fractions, preload and postload glucose levels, smoking exposure, and socioeconomic status) was available. A
noninvasive ultrasound technique was used to assess the ability of the brachial artery to dilate in response to
increased blood flow (induced by forearm cuff occlusion and release), an endothelium-dependent response. Birth
weight showed a significant, graded, positive association with flow-mediated dilation (0.027 mm/kg; 95% CI, 0.003
to 0.051 mm/kg; P=.02). Childhood cardiovascular risk factors (blood pressure, total and LDL cholesterol, and
salivary cotinine level) showed no relation with flow-mediated dilation, but HDL cholesterol level was inversely
related (-0.067 mm/mmol; 95% CI, -0.021 to -0.113 mm/mmol; P=.005). The relation between birth weight and
flow-mediated dilation was not affected <local>by <head>adjustment</head></local> for childhood body build,
parity, cardiovascular risk factors, social class, or ethnicity. CONCLUSIONS: Low birth weight is associated with
impaired endothelial function in childhood, a key early event in atherogenesis. Growth in utero may be associated
with long-term changes in vascular function that are manifest by the first decade of life and that may influence the
long-term risk of cardiovascular disease.
   </context>
  </instance>
  .....
  .....
 </lexelt>
</corpus>
```

Sentence context in SENSEVAL2 format.

```
<corpus lang='en'>
 <lexelt item="adjustment">
  <instance id="9337195.ab.7" pmid="9337195" alias="adjustment">
   <answer instance="9337195.ab.7" senseid="M2"/>
   <context>
    The relation between birth weight and flow-mediated dilation was not affected <local>by <head>adjustment
</head></local> for childhood body build, parity, cardiovascular risk factors, social class, or ethnicity.
   </context>
  </instance>
  .....
  .....
 </lexelt>
</corpus>
```

Fig. 2. SENSEVAL-2 formatted abstract and sentence contexts for the word *adjustment*.

separated by one or more other words in between them. We limit the number of words between the two words of a bigram using a widow size parameter that ranges from two to five. A window size of two means that the words in the bigram have to be adjacent, without any other words in between them and a window size of five means that there can be up to three other words in between the two words of the bigram. Our stop list of pre-defined functional words operates as a disjunctive stop list when filtering out bigrams – even if one of the words in the bigram is a stop word, the bigram is discarded. We then apply the frequency cut-off criteria ranging from two to five as in the case of unigrams. Additionally, we use the log likelihood measure to identify bigrams that occur together more often than by chance. The two acceptance criteria that we use are log likelihood scores of 3.841 and 6.684. These score values indicate that the bigrams are significant (and not random independent co-occurrences) with 95 percent and 99 percent confidence respectively.

5 Evaluation

Our evaluation of SVMs is based on their comparative performance with respect to the other machine learning algorithms. We report our results in terms of the standard measure of *accuracy*, which is the percentage of correctly classified instances. The baseline performance for our evaluation is provided by the ZeroR majority classifier in Weka. A majority classifier is based on the simple rule of assigning the most frequent sense to all the instances. With reference to Table 1, a majority classifier for the word *adjustment* will assign the sense *M2* to all the instances, yielding an accuracy of 62 percent. These majority classifier accuracy values for each of the words serve as the baseline performance for our experiments.

For a given word, we consider a trained model of a classifier significant in performance only if its accuracy is at least five percentage points better than the accuracy of the majority classifier for that word. Given a best classifier for some word, we consider any other significant classifier within three percentage points accuracy of the top classifier to be among the best classifiers. For example if the best classifier for the word *adjustment* yields 75 percent accuracy, then any other classifier having accuracy in the range of 72 to 75 percent will also be counted as a top classifier for *adjustment*.

Our data is not separated into training and test instances. We therefore use the cross-validation mechanism available in Weka for testing the performance of various methods. For all of our experiments we have used ten-fold cross-validation.

6 Results

Table 3 shows high level results for the entire dataset in terms of the number of words for which a particular classifier achieved the highest accuracy. The second column shows the numbers for abstract context and the third column for the

Table 3. High level results for the entire dataset: For each of the evaluated classifiers we have the number of words (out of 50) for which they were the best.

Classifier	**Number of words for which it is best**	
	Abstract Context	**Sentence Context**
SMO	31	33
NB	27	25
ABM1	19	19
DT	17	17
PART	14	16
ZeroR	11	11

sentence context. In all of our result tables, SMO refers to the SVM classifier that we use, NB to the naïve Bayes classifier, DT to the decision tree classifier, ABM1 to the boosting algorithm classifier, PART to the decision list classifier and ZeroR to the majority classifier in Weka.

For all of our following results, we exclude 20 words from the collection where none of the classifiers could achieve at least five percentage point accuracy improvement over the majority classifier, in abstract context. The excluded words are - *association, cold, condition, energy, extraction, failure, fluid, frequency, ganglion, glucose, inhibition, pathology, pressure, reduction, resistance, secretion, single, surgery, transient* and *transport*. We believe that in these cases the other classifiers could not achieve considerable improvement over the majority classifier since most of these words have a majority sense that exceeds 80 percent and therefore have a very skewed distribution which provides significantly fewer instances of senses other than the majority sense. Although the word *failure* does not have an overly skewed distribution, it has a very high number of instances belonging to the sense *None*, which might have degraded the performance of classifiers as discussed earlier.

The best classifiers for every word are selected based on the criteria mentioned in the evaluation section. Table 4 shows the classifiers and the words for which they were the best classifiers in abstract and sentence context. The numbers shown in the third column count the total number of words that a classifier was best for, whereas the number of different models that performed equally well for a given word are shown in parenthesis after each word. For example, in the table the number of words for which SVMs were the best classifiers is 20, and 4 different SVM models performed the best for the word *adjustment*.

Excluding the 20 words skipped for result analysis, there were 60 sets of ambiguous instances of the 30 significant words – two sets per word, one set consisting of sentence contexts and the second consisting of abstract contexts. Table 5 lists the top three models for every classifier, their significant features and the number of sets (out of 60) for which they performed the best. **'U'** in the feature column indicates unigram features and the cut-off column specifies the frequency cut-off applied during feature selection. As seen in the table, all the models that proved best are those trained on unigram features. This emphasizes the conventional wisdom that more features can help train a better classifier,

Table 4. Overall Results: A list of all the evaluated classifiers and the significant words for which they were the best. The number inside parenthesis after every word is the count of distinct classifier models that performed equally well for the given word.

Abstract		
Classifier	**Words**	**Word count**
SMO	adjustment(4) culture(2) degree(3) determination(2) discharge(3) fat(3) immunosuppression(2) implantation(3) lead(3) man(4) mole(3) mosaic(3) nutrition(4) radiation(3) repair(3) scale(2) sex(3) ultrasound(2) weight(3) white(3)	20
NB	adjustment(4) blood_pressure(3) depression(2) discharge(3) evaluation(2) fat(3) fit(2) growth(2) immunosuppression(2) implantation(3) japanese(3) lead(3) mosaic(3) radiation(3) repair(3) scale(2) sensitivity(4) variation(2) white(3)	19
DT	adjustment(4) culture(2) degree(3) immunosuppression(2) man(4) mole(3) radiation(3) scale(2) sex(3) strains(3) support(3) ultrasound(2)	12
ABM1	culture(2) degree(3) fit(2) implantation(3) man(4) mole(3) nutrition(4) scale(2) ultrasound(2) variation(2)	10
PART	blood_pressure(3) culture(2) degree(3) mole(3) nutrition(4) radiation(3) scale(2) sex(3) strains(3) ultrasound(2)	10
Sentence		
Classifier	**Words**	**Word count**
SMO	adjustment(4) culture(2) degree(3) discharge(3) fat(3) fit(2) immunosuppression(2) implantation(3) japanese(3) lead(3) man(4) mole(3) mosaic(3) nutrition(4) repair(3) scale(2) sensitivity(4) sex(3) weight(3) white(3)	20
NB	blood_pressure(3) culture(2) degree(3) discharge(3) evaluation(2) fat(3) fit(2) growth(2) immunosuppression(2) implantation(3) japanese(3) lead(3) man(4) mole(3) mosaic(3) radiation(3) scale(2) sensitivity(4) white(3)	19
PART	blood_pressure(3) culture(2) discharge(3) evaluation(2) fat(3) fit(2) lead(3) man(4) mole(3) nutrition(4) scale(2) white(3)	12
DT	blood_pressure(3) culture(2) discharge(3) fat(3) fit(2) lead(3) man(4) mole(3) nutrition(4) radiation(3) scale(2)	11
ABM1	culture(2) degree(3) fat(3) fit(2) scale(2)	5

which was specially true in the case of naïve Bayes and SVM classifiers. In the case of other classifiers, the performance was unfavorably affected due to a large number of features, reducing the number of sets for which they performed best.

Table 6 shows the best classifiers and their accuracy for each significant word in the abstract and sentence contexts. We also show the average accuracy value

Table 5. Individual Classifier Results: The top three classifier models in each category, their significant feature selection criteria and results out of 60 instances.

Classifier	Feature	Cut-off	Best for # (out of 60)
SMO-1	U	4	27
SMO-2	U	5	26
SMO-3	U	3	25
NB-1	U	3	22
NB-2	U	4	21
NB-3	U	5	21
DT-1	U	5	14
DT-2	U	4	14
DT-3	U	3	13
ABM1-1	U	5	14
ABM1-2	U	3	14
ABM1-3	U	4	14
PART-1	U	2	10
PART-2	U	4	10
PART-3	U	5	9

and standard deviation of all classifiers for each word in both contexts, in the fifth and eighth columns. The second column lists the accuracy values of the majority classifier for each word. We can observe that SVMs performed well for both sentence and abstract contexts. In most cases where SVMs were the best for the abstract as well as the sentence context (*immunosuppression, implantation, lead, sex, ultrasound* and *weight*), their performance was better in the abstract context. Exceptions to this were the words *man* and *mole* where SVMs performed better in the smaller sentence context. This suggests that SVMs can perform well not only when more features are present but also in the presence of lesser but indicative features. In particular, a comparison of best SVM classifier results for all the words reveals that in 12 cases out of 50, the results in sentence context outperformed those in abstract context. Table 7 lists these words and the accuracy of SVMs in abstract versus sentence contexts. The performance in sentence context is strikingly high for *nutrition* and *blood_pressure* with improvements of 11 and 9 percentage points respectively. For other words like *degree, fit, japanese, man, mole* and *pathology* the improvement is 3 or 4 percentage points which is still significant.

Table 8 shows the comparison of our results with that of Liu et al. [10] and Leroy and Rindflesch [9]. Note that the set of words evaluated by Liu et al. and Leroy and Rindflesch is different from the 30 words that we have analyzed. The table however includes comparison with the exact set of words used by both of them. Even with a limited feature set of unigrams or bigrams in context, the SVM classifier was able to outperform results from Liu et al. in 11 cases out of 22 – 5 times in abstract context, 4 times in sentence context and 2 times in both contexts. SVM accuracy in either the abstract, sentence or both the contexts

Table 6. Results for significant words: Shown are the best classifiers, their accuracy for each word in the abstract and sentence contexts and the average accuracy of all classifiers for every word, with standard deviation.

		Abstract			Sentence		
Word	**Maj.**	**Classifier**	**Acc.**	**Avg.**	**Classifier**	**Acc.**	**Avg.**
adjustment	62	DT,NB	72	65.86±3.61	**SMO**	70	64.54±2.44
blood_pressure	54	NB	61	52.18±5.10	NB	66	55.22±3.93
culture	89	DT,ABM1, PART	99	89.51±2.92	**SMO**	97	89.26±2.04
degree	63	ABM1	92	63.77±9.42	**SMO**,NB ABM1	92	65.97±8.62
depression	85	NB	90	84.23±1.57	**SMO**	87	84.90±1.28
determination	79	**SMO**	85	77.22±2.40	ABM1	80	78.58±1.34
discharge	74	**SMO**,NB	95	76.09±4.84	DT	83	74.45±1.79
evaluation	50	NB	75	54.71±5.91	PART	67	50.80±3.65
fat	71	NB	87	80.06±3.59	NB	82	79.13±2.08
fit	82	ABM1,NB	90	82.41±1.81	NB	91	82.87±2.34
growth	63	NB	75	64.22±4.57	NB	70	61.74±1.97
immunosuppression	59	**SMO**	80	64.70±7.11	NB,**SMO**	72	59.42±3.13
implantation	81	**SMO**	94	87.52±1.94	**SMO**,NB	86	81.97±1.29
japanese	73	NB	78	73.22±2.10	**SMO**	81	74.39±1.96
lead	71	**SMO**	89	82.87±3.52	**SMO**,NB	83	72.28±2.48
man	58	**SMO**	89	77.86±5.44	**SMO**	92	81.45±3.21
mole	83	**SMO**	95	87.66±3.17	**SMO**	98	91.60±2.23
mosaic	52	**SMO**	87	66.05±7.39	NB	79	61.12±5.48
nutrition	45	ABM1	55	43.34±3.95	DT	65	48.83±4.61
radiation	61	**SMO**,NB	82	74.72±4.70	NB	74	62.26±2.60
repair	52	NB	88	71.70±8.74	**SMO**	72	56.88±4.62
scale	65	NB,ABM1	82	79.46±3.39	**SMO**	80	73.43±6.19
sensitivity	48	NB	92	65.36±11.88	NB	78	54.81±5.63
sex	80	**SMO**	88	84.02±2.01	**SMO**	85	80.34±1.47
strains	92	PART,DT	97	91.79±1.29	PART	92	91.51±0.79
support	90	DT	95	91.39±1.80	ABM1	93	89.98±0.56
ultrasound	84	**SMO**	92	87.20±1.84	**SMO**	85	83.41±1.35
variation	80	ABM1	92	84.17±3.18	NB	83	79.71±1.24
weight	47	**SMO**	83	57.85±10.11	**SMO**	80	49.29±9.29
white	49	NB	80	66.85±8.64	NB,**SMO**	72	58.25±7.24

was better than the best results by Leroy and Rindflesch for all words except *scale*. It is interesting that without making use of BOW features they achieved better performance in this case. This suggests that augmenting BOW features with their feature set might enhance performance further.

Table 9 shows the average accuracy of the all the classifier models that we evaluated for the 30 significant words, along with the standard deviation values. Except for the ZeroR majority classifier, 180 test cases (36 per classifier) were run for each of the 30 significant words. A total of 1080 (36 x 30) tests were run for each classifier. Out of the 1080 tests, 174 test cases in sentence context

Table 7. Comparison of SVM classifier accuracy where results in sentence context outperformed those in abstract context.

Word	Majority	Abstract	Sentence
blood_pressure	54	53	62
condition	90	90	91
culture	89	96	97
degree	63	89	92
depression	85	86	87
fit	82	86	90
japanese	73	77	81
man	58	89	92
mole	83	95	98
nutrition	45	52	63
pathology	85	85	88
reduction	89	91	93

using bigram features could not identify even a single significant bigram, given the small size of the sentence context. As a result, the input to the classifiers in these cases did not have any features. While all other classifiers defaulted to the majority sense in these test cases, the boosting classifier failed because of the way it is implementated in Weka. Therefore for such tests we assumed that the boosting classifier would also ideally revert to the majority sense and then calculated the average accuracy value for it. On average, all of the classifiers we evaluated performed better in the abstract context. SVMs showed an average improvement of 6 percentage points over the majority classifier and proved to be the best among all the classifiers that we evaluated.

7 Future Work

The experiments that we have performed so far can be fine tuned to achieve better results. We organize the future work into two categories.

(a) *Feature engineering* : This involves identifying a better set of features from the data. Incorporating stemming approach to remove morphological variations of the ambiguous words as well as the features is one of the first steps. We hope to achieve a more concise and richer feature set via this approach, which we believe will improve SVM performance. POS tags of words in a small window size around the ambiguous words are very useful features, as demonstrated in [8] and [11]. We would like to make use of such POS features along with syntactic relationships wherever possible. Unlike Liu et al. [10] we have not considered the orientation and distance information of unigrams. Including this information as features should boost the performance of SVMs and also other classifiers in general. Although the existing methods using conventional word sense disambiguation features have performed well, it will be interesting to explore any domain specific features that apply to the medical text. Finally, the data that we have used is a fairly small test collection. With large amounts of data be-

Table 8. Comparison with results of Liu et al. [10] and Leroy and Rindflesch [9]. The table shows the comparative performance of SMO with the best results from Liu et al. and Leroy and Rindflesch and also shows in the last column the best results obtained in our experiments and the classifiers that obtained them. For the last two columns anything to the left of the separator "/" is for abstract context and anything to the right of it is for sentence context. Numbers in bold highlight cases where SMO outperformed best results from Liu et al.

Word	**Majority**	**Best Result**			
		Liu	**Leroy**	**SMO**	**Overall (for our experiments)**
adjustment	62	-	62	71/70	72/70 (NB/SMO)
blood_pressure	54	-	56	53/62	61/66 (NB/NB)
cold	86	90.9	-	90/88	90/89 (SMO/ABM1)
degree	63	98.2	70	89/92	92/92 (ABM1/NB,SMO,ABM1)
depression	85	88.8	-	86/87	90/87 (NB/NB,SMO)
discharge	74	90.8	-	**95**/82	95/83 (NB,SMO/DT)
evaluation	50	-	57	69/62	75/67 (NB/PART)
extraction	82	89.7	-	84/84	84/85 (NB,SMO/DT)
fat	71	85.9	-	84/80	87/82 (NB/NB)
growth	63	72.2	63	71/63	75/70 (NB/NB)
immunosuppression	59	-	67	80/72	80/72 (SMO/NB,SMO)
implantation	81	90.0	-	**94**/86	94/86 (SMO/NB,SMO)
japanese	73	79.8	-	77/**81**	78/81 (NB/SMO)
lead	71	91.0	-	89/83	89/83 (SMO/NB,SMO)
man	58	91.0	80	89/**92**	89/92 (SMO/SMO)
mole	83	91.1	-	**95/98**	95/98 (SMO/SMO)
mosaic	52	87.8	69	87/77	87/79 (SMO/NB)
nutrition	45	58.1	53	52/**63**	55/65 (ABM1/DT)
pathology	85	88.2	-	85/88	86/88 (ABM1/SMO,ABM1)
radiation	61	-	72	82/69	82/74 (SMO/NB)
reduction	89	91.0	-	91/**93**	91/93 (SMO,ABM1/SMO,PART)
repair	52	76.1	81	**87**/72	88/72 (NB/SMO)
scale	65	90.9	84	81/80	82/80 (NB,ABM1/SMO)
sensitivity	48	-	70	88/76	92/78 (NB/NB)
sex	80	89.9	-	88/85	88/85 (SMO/SMO)
ultrasound	84	87.8	-	**92**/85	92/85 (SMO/SMO)
weight	47	78.0	71	**83/80**	83/80 (SMO/SMO)
white	49	75.6	62	**79**/72	80/72 (NB/NB,SMO)

ing produced in different medical institutions, a larger data collection could be used for identifying better features. However, such large data collections may not include manually sense-tagged instances, which introduces the possibility of employing semi-supervised approaches where in unsupervised methods are used for automatic training data generation and then supervised methods are trained using this data.

(b) *SVM tuning* : One of the possibilities under this category is to tune the parameters of the SMO classifier. We have only experimented with the default linear kernel. Testing polynomial kernels and Radial Basis Function kernels for

Table 9. Average accuracy (with standard deviation) for all the classifiers that that we evaluated, over all the 30 significant words.

Context	Classifier	Accuracy (%)	# Tests
Abstract	SMO	76.26±12.93	1080
	NB	75.03±11.53	1080
	DT	74.62±13.12	1080
	PART	73.71±13.69	1080
	ABM1	71.72±15.26	1080
	ZeroR	68.07±14.64	30
Sentence	SMO	71.90±13.65	1080
	NB	71.56±13.17	1080
	PART	71.12±13.93	1080
	DT	71.06±14.14	1080
	ABM1	70.47±14.53	1080
	ZeroR	68.07±14.64	30
Overall	SMO	74.08±13.47	2160
	NB	73.29±12.50	2160
	DT	72.84±13.75	2160
	PART	72.42±13.87	2160
	ABM1	71.09±14.93	2160
	ZeroR	68.07±14.51	60

SVMs can significantly improve the accuracy of our results. Additionally, the idea of domain kernels for word sense disambiguation [6] can be explored as a part of our future experiments.

8 Conclusion

The results from our experiments so far indicate that Support Vector Machines are promising candidates for further research in supervised word sense disambiguation in the medical domain. They outperformed other classifiers in most of our experiments and gave their best performance with unigram features selected using a frequency cut-off of four. When SVMs were not the best classifiers for a word, they were at least close to the best classifiers within a small margin of two to three percentage point accuracy – suggesting that after tuning the feature set and classifier options, they can perform better.

9 Acknowledgments

Dr. Pedersen has been partially supported in carrying out this research by a National Science Foundation Faculty Early CAREER Development Award (#0092784).

References

1. Baker C. F., Fillmore C. J., and Lowe J.B.: The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics. Morgan Kaufmann Publishers, San Francisco, CA. (1998) 86–90
2. Banerjee S. and Pedersen T.: The Design, Implementation and Use of the Ngram Statistics Package. Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City (2003)
3. Cabezas C., Resnik P. and Stevens J.: Supervised Sense Tagging using Support Vector Machines. Proceedings of SENSEVAL-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France. SIGLEX, Association for Computational Linguistics (2001) 59–62
4. Frank E. and Witten I.: Generating Accurate Rule Sets Without Global Optimization. Proceedings of the Fifteenth International Conference on Machine Learning, Madison, Wisconsin. Shavlik, J. (editor). Morgan Kaufmann Publishers, San Francisco, CA. (1998) 144–151
5. Freund Y. and Schapire R.: Experiments with a new boosting algorithm. Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA. (1996) 148–156
6. Gliozzo A., Giuliano C. and Strapparava C.: Domain Kernels for Word Sense Disambiguation. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor. (2005) 403–410
7. Hastie T. and Tibshirani R.: Classification by Pairwise Coupling. The Annals of Statistics, Vol. 26, No. 2. (1998) 451–471
8. Lee Y. K., Ng H. T. and Chia T. K.: Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain. Association for Computational Linguistics (2004)
9. Leroy G. and Rindflesch T. C.: Using Symbolic Knowledge in the UMLS to Disambiguate Words in Small Datasets with a Naïve Bayes Classifier. MEDINFO 2004, San Francisco. Fieschi M. et al. (editors), IOS Press, Amsterdam
10. Liu H., Teller V. and Friedman C.: A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation. Journal of the American Medical Informatics Association (2004)
11. Ngai G., Wu D., Carpuat M., Wang C. and Wang C.: Semantic Role Labeling with Boosting, SVMs, Maximum Entropy, SNOW, and Decision Lists. SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain. Association for Computational Linguistics (2004)
12. Platt J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in kernel methods - support vector learning. B. Schölkopf, C. Burges, and A. Smola (editors), MIT Press. (1998)
13. Schapire R.: The Boosting Approach To Machine Learning: An Overview. MSRI Workshop on Nonlinear Estimation and Classification. (2002)
14. Vapnik V.: The Nature of Statistical Learning Theory. Springer. (1995)
15. Weeber M., Mork J. and Aronson A.: Developing a Test Collection for Biomedical Word Sense Disambiguation. Proceedings of AMIA Symposium (2001) 746–750
16. Witten I. and Frank E.: Data Mining: Practical machine learning tools with Java implementation. Morgan-Kaufmann (2000)