Domain Kernels for Word Sense Disambiguation

Alfio Gliozzo and **Claudio Giuliano** and **Carlo Strapparava**
ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica
I-38050, Trento, ITALY
{gliozzo,giuliano,strappa}@itc.it

Abstract

In this paper we present a supervised Word Sense Disambiguation methodology, that exploits kernel methods to model sense distinctions. In particular a combination of kernel functions is adopted to estimate independently both *syntagmatic* and *domain* similarity. We defined a kernel function, namely the Domain Kernel, that allowed us to plug "external knowledge" into the supervised learning process. External knowledge is acquired from unlabeled data in a totally unsupervised way, and it is represented by means of Domain Models. We evaluated our methodology on several lexical sample tasks in different languages, outperforming significantly the state-of-the-art for each of them, while reducing the amount of labeled training data required for learning.

1 Introduction

The main limitation of many supervised approaches for Natural Language Processing (NLP) is the lack of available annotated training data. This problem is known as the Knowledge Acquisition Bottleneck.

To reach high accuracy, state-of-the-art systems for Word Sense Disambiguation (WSD) are designed according to a supervised learning framework, in which the disambiguation of each word in the lexicon is performed by constructing a different classifier. A large set of sense tagged examples is then required to train each classifier. This methodology is called *word expert* approach (Small, 1980; Yarowsky and Florian, 2002). However this is clearly unfeasible for *all-words* WSD tasks, in which all the words of an open text should be disambiguated.

On the other hand, the word expert approach works very well for *lexical sample* WSD tasks (i.e. tasks in which it is required to disambiguate only those words for which enough training data is provided). As the original rationale of the lexical sample tasks was to define a clear experimental settings to enhance the comprehension of WSD, they should be considered as *preceding* exercises to all-words tasks. However this is not the actual case. Algorithms designed for lexical sample WSD are often based on pure supervision and hence "data hungry".

We think that lexical sample WSD should regain its original *explorative* role and possibly use a minimal amount of training data, exploiting instead external knowledge acquired in an unsupervised way to reach the actual state-of-the-art performance.

By the way, minimal supervision is the basis of state-of-the-art systems for all-words tasks (e.g. (Mihalcea and Faruque, 2004; Decadt et al., 2004)), that are trained on small sense tagged corpora (e.g. SemCor), in which few examples for a subset of the ambiguous words in the lexicon can be found. Thus improving the performance of WSD systems with few learning examples is a fundamental step towards the direction of designing a WSD system that works well on real texts.

In addition, it is a common opinion that the performance of state-of-the-art WSD systems is not satisfactory from an applicative point of view yet.

To achieve these goals we identified two promising research directions:

1. Modeling independently domain and syntagmatic aspects of sense distinction, to improve the feature representation of sense tagged examples (Gliozzo et al., 2004).

2. Leveraging external knowledge acquired from unlabeled corpora.

The first direction is motivated by the linguistic assumption that syntagmatic and domain (associative) relations are both crucial to represent sense distictions, while they are basically originated by very different phenomena. Syntagmatic relations hold among words that are typically located close to each other in the same sentence in a given temporal order, while domain relations hold among words that are typically used in the same semantic domain (i.e. in texts having similar topics (Gliozzo et al., 2004)). Their different nature suggests to adopt different learning strategies to detect them.

Regarding the second direction, external knowledge would be required to help WSD algorithms to better generalize over the data available for training. On the other hand, most of the state-of-the-art supervised approaches to WSD are still completely based on "internal" information only (i.e. the only information available to the training algorithm is the set of manually annotated examples). For example, in the Senseval-3 evaluation exercise (Mihalcea and Edmonds, 2004) many lexical sample tasks were provided, beyond the usual labeled training data, with a large set of unlabeled data. However, at our knowledge, none of the participants exploited this unlabeled material. Exploring this direction is the main focus of this paper. In particular we acquire a Domain Model (DM) for the lexicon (i.e. a lexical resource representing domain associations among terms), and we exploit this information inside our supervised WSD algorithm. DMs can be automatically induced from unlabeled corpora, allowing the portability of the methodology among languages.

We identified kernel methods as a viable framework in which to implement the assumptions above (Strapparava et al., 2004).

Exploiting the properties of kernels, we have defined independently a set of domain and syntagmatic kernels and we combined them in order to define a complete kernel for WSD. The domain kernels estimate the (domain) similarity (Magnini et al., 2002) among contexts, while the syntagmatic kernels evaluate the similarity among collocations.

We will demonstrate that using DMs induced from unlabeled corpora is a feasible strategy to increase the generalization capability of the WSD algorithm. Our system far outperforms the state-of-the-art systems in all the tasks in which it has been tested. Moreover, a comparative analysis of the learning curves shows that the use of DMs allows us to remarkably reduce the amount of sense-tagged examples, opening new scenarios to develop systems for all-words tasks with minimal supervision.

The paper is structured as follows. Section 2 introduces the notion of Domain Model. In particular an automatic acquisition technique based on Latent Semantic Analysis (LSA) is described. In Section 3 we present a WSD system based on a combination of kernels. In particular we define a Domain Kernel (see Section 3.1) and a Syntagmatic Kernel (see Section 3.2), to model separately syntagmatic and domain aspects. In Section 4 our WSD system is evaluated in the Senseval-3 English, Italian, Spanish and Catalan lexical sample tasks.

2 Domain Models

The simplest methodology to estimate the similarity among the topics of two texts is to represent them by means of vectors in the Vector Space Model (VSM), and to exploit the cosine similarity. More formally, let $C = \{t_1, t_2, \ldots, t_n\}$ be a corpus, let $V = \{w_1, w_2, \ldots, w_k\}$ be its vocabulary, let $\mathbf{T}$ be the $k \times n$ term-by-document matrix representing C, such that $\mathbf{t_{i,j}}$ is the frequency of word w_i into the text t_j. The VSM is a k-dimensional space $\mathbb{R}^k$, in which the text $t_j \in C$ is represented by means of the vector $\vec{t_j}$ such that the i^{th} component of $\vec{t_j}$ is $\mathbf{t_{i,j}}$. The similarity among two texts in the VSM is estimated by computing the cosine among them.

However this approach does not deal well with lexical variability and ambiguity. For example the two sentences "*he is affected by AIDS*" and "*HIV is a virus*" do not have any words in common. In the

VSM their similarity is zero because they have orthogonal vectors, even if the concepts they express are very closely related. On the other hand, the similarity between the two sentences "*the laptop has been infected by a **virus***" and "*HIV is a **virus***" would turn out very high, due to the ambiguity of the word `virus`.

To overcome this problem we introduce the notion of *Domain Model* (DM), and we show how to use it in order to define a *domain VSM* in which texts and terms are represented in a uniform way.

A DM is composed by soft clusters of terms. Each cluster represents a semantic domain, i.e. a set of terms that often co-occur in texts having similar topics. A DM is represented by a $k \times k'$ rectangular matrix $\mathbf{D}$, containing the degree of association among terms and domains, as illustrated in Table 1.

	MEDICINE	COMPUTER_SCIENCE
HIV	1	0
AIDS	1	0
virus	0.5	0.5
laptop	0	1

Table 1: Example of Domain Matrix

DMs can be used to describe lexical ambiguity and variability. Lexical ambiguity is represented by associating one term to more than one domain, while variability is represented by associating different terms to the same domain. For example the term `virus` is associated to both the domain COMPUTER_SCIENCE and the domain MEDICINE (ambiguity) while the domain MEDICINE is associated to both the terms `AIDS` and `HIV` (variability).

More formally, let $\mathcal{D} = \{D_1, D_2, ..., D_{k'}\}$ be a set of domains, such that $k' \ll k$. A DM is fully defined by a $k \times k'$ *domain matrix* $\mathbf{D}$ representing in each cell $\mathbf{d_{i,z}}$ the *domain relevance* of term w_i with respect to the domain D_z. The domain matrix $\mathbf{D}$ is used to define a function $\mathcal{D} : \mathbb{R}^k \rightarrow \mathbb{R}^{k'}$, that maps the vectors $\vec{t_j}$ expressed into the classical VSM, into the vectors $\vec{t'_j}$ in the domain VSM. $\mathcal{D}$ is defined by[1]

$$\mathcal{D}(\vec{t_j}) = \vec{t_j}(\mathbf{I^{IDF}D}) = \vec{t'_j} \quad (1)$$

where $\mathbf{I^{IDF}}$ is a $k \times k$ diagonal matrix such that $i^{IDF}_{i,i} = IDF(w_i)$, $\vec{t_j}$ is represented as a row vector, and $IDF(w_i)$ is the *Inverse Document Frequency* of w_i.

Vectors in the domain VSM are called Domain Vectors (DVs). DVs for texts are estimated by exploiting the formula 1, while the DV $\vec{w'_i}$, corresponding to the word $w_i \in V$ is the i^{th} row of the domain matrix $\mathbf{D}$. To be a valid domain matrix such vectors should be normalized (i,e. $\langle \vec{w'_i}, \vec{w'_i} \rangle = 1$).

In the Domain VSM the similarity among DVs is estimated by taking into account second order relations among terms. For example the similarity of the two sentences "*He is affected by AIDS*" and "*HIV is a virus*" is very high, because the terms `AIDS`, `HIV` and `virus` are highly associated to the domain MEDICINE.

A DM can be estimated from hand made lexical resources such as WORDNET DOMAINS (Magnini and Cavaglià, 2000), or by performing a term clustering process on a large corpus. We think that the second methodology is more attractive, because it allows us to automatically acquire DMs for different languages.

In this work we propose the use of Latent Semantic Analysis (LSA) to induce DMs from corpora. LSA is an unsupervised technique for estimating the similarity among texts and terms in a corpus. LSA is performed by means of a Singular Value Decomposition (SVD) of the term-by-document matrix $\mathbf{T}$ describing the corpus. The SVD algorithm can be exploited to acquire a domain matrix $\mathbf{D}$ from a large corpus C in a totally unsupervised way. SVD decomposes the term-by-document matrix $\mathbf{T}$ into three matrixes $\mathbf{T} \simeq \mathbf{V\Sigma_{k'}U}^T$ where $\mathbf{\Sigma_{k'}}$ is the diagonal $k \times k$ matrix containing the highest $k' \ll k$ eigenvalues of $\mathbf{T}$, and all the remaining elements set to 0. The parameter k' is the dimensionality of the Domain VSM and can be fixed in advance[2]. Under this setting we define the domain matrix $\mathbf{D_{LSA}}$ as

$$\mathbf{D_{LSA}} = \mathbf{I^N V}\sqrt{\mathbf{\Sigma_{k'}}} \quad (2)$$

where $\mathbf{I^N}$ is a diagonal matrix such that $\mathbf{i^N_{i,i}} = \frac{1}{\sqrt{\langle \vec{w'_i}, \vec{w'_i} \rangle}}$, $\vec{w'_i}$ is the i^{th} row of the matrix $\mathbf{V}\sqrt{\mathbf{\Sigma_{k'}}}$.[3]

[1] In (Wong et al., 1985) the formula 1 is used to define a Generalized Vector Space Model, of which the Domain VSM is a particular instance.

[2] It is not clear how to choose the right dimensionality. In our experiments we used 50 dimensions.

[3] When $\mathbf{D_{LSA}}$ is substituted in Equation 1 the Domain VSM

3 Kernel Methods for WSD

In the introduction we discussed two promising directions for improving the performance of a supervised disambiguation system. In this section we show how these requirements can be efficiently implemented in a natural and elegant way by using kernel methods.

The basic idea behind kernel methods is to embed the data into a suitable feature space $\mathcal{F}$ via a mapping function $\phi : \mathcal{X} \rightarrow \mathcal{F}$, and then use a linear algorithm for discovering nonlinear patterns. Instead of using the explicit mapping ϕ, we can use a kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, that corresponds to the inner product in a feature space which is, in general, different from the input space.

Kernel methods allow us to build a modular system, as the kernel function acts as an interface between the data and the learning algorithm. Thus the kernel function becomes the only domain specific module of the system, while the learning algorithm is a general purpose component. Potentially any kernel function can work with any kernel-based algorithm. In our system we use Support Vector Machines (Cristianini and Shawe-Taylor, 2000).

Exploiting the properties of the kernel functions, it is possible to define the kernel combination schema as

$$K_C(x_i, x_j) = \sum_{l=1}^{n} \frac{K_l(x_i, x_j)}{\sqrt{K_l(x_j, x_j) K_l(x_i, x_i)}} \quad (3)$$

Our WSD system is then defined as combination of n basic kernels. Each kernel adds some additional dimensions to the feature space. In particular, we have defined two families of kernels: *Domain* and *Syntagmatic* kernels. The former is composed by both the Domain Kernel (K_D) and the Bag-of-Words kernel (K_{BoW}), that captures domain aspects (see Section 3.1). The latter captures the syntagmatic aspects of sense distinction and it is composed by two kernels: the collocation kernel (K_{Coll}) and the Part of Speech kernel (K_{PoS}) (see Section 3.2). The WSD kernels (K'_{WSD} and K_{WSD}) are then defined by combining them (see Section 3.3).

3.1 Domain Kernels

In (Magnini et al., 2002), it has been claimed that knowing the domain of the text in which the word is located is a crucial information for WSD. For example the (domain) polysemy among the COMPUTER_SCIENCE and the MEDICINE senses of the word `virus` can be solved by simply considering the domain of the context in which it is located.

This assumption can be modeled by defining a kernel that estimates the domain similarity among the contexts of the words to be disambiguated, namely the *Domain Kernel*. The Domain Kernel estimates the similarity among the topics (domains) of two texts, so to capture domain aspects of sense distinction. It is a variation of the Latent Semantic Kernel (Shawe-Taylor and Cristianini, 2004), in which a DM (see Section 2) is exploited to define an explicit mapping $\mathcal{D} : \mathbb{R}^k \rightarrow \mathbb{R}^{k'}$ from the classical VSM into the Domain VSM. The Domain Kernel is defined by

$$K_D(t_i, t_j) = \frac{\langle \mathcal{D}(t_i), \mathcal{D}(t_j) \rangle}{\sqrt{\langle \mathcal{D}(t_j), \mathcal{D}(t_j) \rangle \langle \mathcal{D}(t_i), \mathcal{D}(t_i) \rangle}} \quad (4)$$

where $\mathcal{D}$ is the Domain Mapping defined in equation 1. Thus the Domain Kernel requires a Domain Matrix $\mathbf{D}$. For our experiments we acquire the matrix $\mathbf{D_{LSA}}$, described in equation 2, from a generic collection of unlabeled documents, as explained in Section 2.

A more traditional approach to detect topic (domain) similarity is to extract Bag-of-Words (BoW) features from a large window of text around the word to be disambiguated. The BoW kernel, denoted by K_{BoW}, is a particular case of the Domain Kernel, in which $\mathbf{D} = \mathbf{I}$, and $\mathbf{I}$ is the identity matrix. The BoW kernel does not require a DM, then it can be applied to the "strictly" supervised settings, in which an external knowledge source is not provided.

3.2 Syntagmatic kernels

Kernel functions are not restricted to operate on vectorial objects $\vec{x} \in \mathbb{R}^k$. In principle kernels can be defined for any kind of object representation, as for

is equivalent to a Latent Semantic Space (Deerwester et al., 1990). The only difference in our formulation is that the vectors representing the terms in the Domain VSM are normalized by the matrix $\mathbf{I^N}$, and then rescaled, according to their IDF value, by matrix $\mathbf{I^{IDF}}$. Note the analogy with the *tf idf* term weighting schema (Salton and McGill, 1983), widely adopted in Information Retrieval.

example sequences and trees. As stated in Section 1, syntagmatic relations hold among words collocated in a particular temporal order, thus they can be modeled by analyzing sequences of words.

We identified the string kernel (or word sequence kernel) (Shawe-Taylor and Cristianini, 2004) as a valid instrument to model our assumptions. The string kernel counts how many times a (non-contiguous) subsequence of symbols u of length n occurs in the input string s, and penalizes non-contiguous occurrences according to the number of gaps they contain (gap-weighted subsequence kernel).

Formally, let V be the vocabulary, the feature space associated with the gap-weighted subsequence kernel of length n is indexed by a set I of subsequences over V of length n. The (explicit) mapping function is defined by

$$\phi_u^n(s) = \sum_{\mathbf{i}:u=s(\mathbf{i})} \lambda^{l(\mathbf{i})}, u \in V^n \tag{5}$$

where $u = s(\mathbf{i})$ is a subsequence of s in the positions given by the tuple $\mathbf{i}$, $l(\mathbf{i})$ is the length spanned by u, and $\lambda \in]0, 1]$ is the decay factor used to penalize non-contiguous subsequences.

The associate gap-weighted subsequence kernel is defined by

$$k^n(s_i, s_j) = \langle \phi^n(s_i), \phi^n(s_j) \rangle = \sum_{u \in V^n} \phi^n(s_i)\phi^n(s_j) \tag{6}$$

We modified the generic definition of the string kernel in order to make it able to recognize collocations in a local window of the word to be disambiguated. In particular we defined two Syntagmatic kernels: the *n-gram* Collocation Kernel and the *n-gram* PoS Kernel. The *n-gram* Collocation kernel K_{Coll}^n is defined as a gap-weighted subsequence kernel applied to sequences of lemmata around the word l_0 to be disambiguated (i.e. l_{-3}, l_{-2}, l_{-1}, l_0, l_{+1}, l_{+2}, l_{+3}). This formulation allows us to estimate the number of common (sparse) subsequences of lemmata (i.e. collocations) between two examples, in order to capture syntagmatic similarity. In analogy we defined the PoS kernel K_{PoS}^n, by setting s to the sequence of PoSs p_{-3}, p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}, where p_0 is the PoS of the word to be disambiguated.

The definition of the gap-weighted subsequence kernel, provided by equation 6, depends on the parameter n, that represents the length of the subsequences analyzed when estimating the similarity among sequences. For example, K_{Coll}^2 allows us to represent the bigrams around the word to be disambiguated in a more flexible way (i.e. bigrams can be sparse). In WSD, typical features are bigrams and trigrams of lemmata and PoSs around the word to be disambiguated, then we defined the Collocation Kernel and the PoS Kernel respectively by equations 7 and 8[4].

$$K_{Coll}(s_i, s_j) = \sum_{l=1}^{p} K_{Coll}^l(s_i, s_j) \tag{7}$$

$$K_{PoS}(s_i, s_j) = \sum_{l=1}^{p} K_{PoS}^l(s_i, s_j) \tag{8}$$

3.3 WSD kernels

In order to show the impact of using Domain Models in the supervised learning process, we defined two WSD kernels, by applying the kernel combination schema described by equation 3. Thus the following WSD kernels are fully specified by the list of the kernels that compose them.

$\mathbf{K_{wsd}}$ composed by K_{Coll}, K_{PoS} and K_{BoW}

$\mathbf{K'_{wsd}}$ composed by K_{Coll}, K_{PoS}, K_{BoW} and K_D

The only difference between the two systems is that K'_{wsd} uses Domain Kernel K_D. K'_{wsd} exploits external knowledge, in contrast to K_{wsd}, whose only available information is the labeled training data.

4 Evaluation and Discussion

In this section we present the performance of our kernel-based algorithms for WSD. The objectives of these experiments are:

- to study the combination of different kernels,
- to understand the benefits of plugging external information using domain models,
- to verify the portability of our methodology among different languages.

[4] The parameters p and λ are optimized by cross-validation. The best results are obtained setting $p = 2$, $\lambda = 0.5$ for K_{Coll} and $\lambda \to 0$ for K_{PoS}.

4.1 WSD tasks

We conducted the experiments on four lexical sample tasks (English, Catalan, Italian and Spanish) of the Senseval-3 competition (Mihalcea and Edmonds, 2004). Table 2 describes the tasks by reporting the number of words to be disambiguated, the mean polysemy, and the dimension of training, test and unlabeled corpora. Note that the organizers of the English task did not provide any unlabeled material. So for English we used a domain model built from a portion of BNC corpus, while for Spanish, Italian and Catalan we acquired DMs from the unlabeled corpora made available by the organizers.

	#w	*pol*	*# train*	*# test*	*# unlab*
Catalan	27	3.11	4469	2253	23935
English	57	6.47	7860	3944	-
Italian	45	6.30	5145	2439	74788
Spanish	46	3.30	8430	4195	61252

Table 2: Dataset descriptions

4.2 Kernel Combination

In this section we present an experiment to empirically study the kernel combination. The basic kernels (i.e. K_{BoW}, K_D, K_{Coll} and K_{PoS}) have been compared to the combined ones (i.e. K_{wsd} and K'_{wsd}) on the English lexical sample task.

The results are reported in Table 3. The results show that combining kernels significantly improves the performance of the system.

	K_D	K_{BoW}	K_{PoS}	K_{Coll}	K_{wsd}	K'_{wsd}
F1	65.5	63.7	62.9	66.7	**69.7**	**73.3**

Table 3: The performance (F1) of each basic kernel and their combination for English lexical sample task.

4.3 Portability and Performance

We evaluated the performance of K'_{wsd} and K_{wsd} on the lexical sample tasks described above. The results are showed in Table 4 and indicate that using DMs allowed K'_{wsd} to significantly outperform K_{wsd}.

In addition, K'_{wsd} turns out the best systems for all the tested Senseval-3 tasks.

Finally, the performance of K'_{wsd} are higher than the human agreement for the English and Spanish tasks[5].

Note that, in order to guarantee an uniform application to any language, we do not use any syntactic information provided by a parser.

4.4 Learning Curves

The Figures 1, 2, 3 and 4 show the learning curves evaluated on K'_{wsd} and K_{wsd} for all the lexical sample tasks.

The learning curves indicate that K'_{wsd} is far superior to K_{wsd} for all the tasks, even with few examples. The result is extremely promising, for it demonstrates that DMs allow to drastically reduce the amount of sense tagged data required for learning. It is worth noting, as reported in Table 5, that K'_{wsd} achieves the same performance of K_{wsd} using about half of the training data.

	% of training
English	54
Catalan	46
Italian	51
Spanish	50

Table 5: Percentage of sense tagged examples required by K'_{wsd} to achieve the same performance of K_{wsd} with full training.

5 Conclusion and Future Works

In this paper we presented a supervised algorithm for WSD, based on a combination of kernel functions. In particular we modeled domain and syntagmatic aspects of sense distinctions by defining respectively domain and syntagmatic kernels. The Domain kernel exploits Domain Models, acquired from "external" untagged corpora, to estimate the similarity among the contexts of the words to be disambiguated. The syntagmatic kernels evaluate the similarity between collocations.

We evaluated our algorithm on several Senseval-3 lexical sample tasks (i.e. English, Spanish, Italian and Catalan) significantly improving the state-ot-the-art for all of them. In addition, the performance

[5] It is not clear if the inter-annotator-agreement can be considerated the upper bound for a WSD system.

	MF	*Agreement*	*BEST*	K_{wsd}	K'_{wsd}	*DM+*
English	55.2	67.3	72.9	69.7	**73.3**	3.6
Catalan	66.3	93.1	85.2	85.2	**89.0**	3.8
Italian	18.0	89.0	53.1	53.1	**61.3**	8.2
Spanish	67.7	85.3	84.2	84.2	**88.2**	4.0

Table 4: Comparative evaluation on the lexical sample tasks. Columns report: the *Most Frequent* baseline, the *inter annotator agreement*, the *F1* of the best system at Senseval-3, the *F1* of K_{wsd}, the *F1* of K'_{wsd}, *DM+* (the improvement due to DM, i.e. $K'_{wsd} - K_{wsd}$).

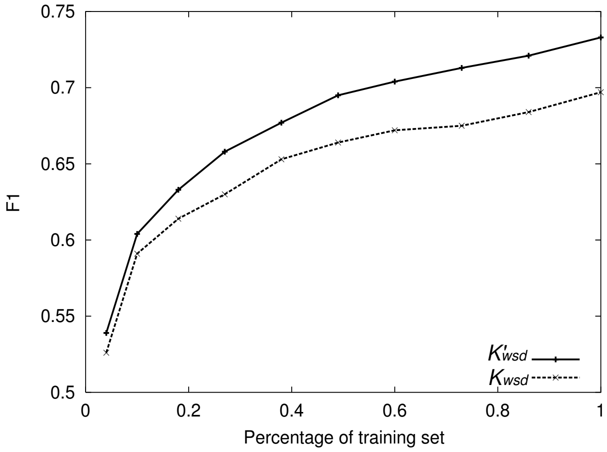

Figure 1: Learning curves for English lexical sample task.

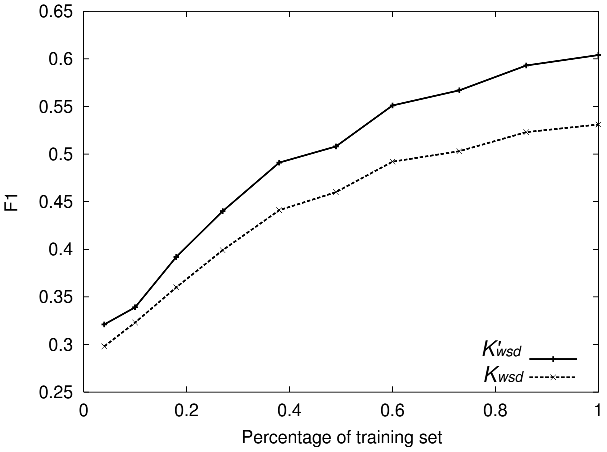

Figure 3: Learning curves for Italian lexical sample task.

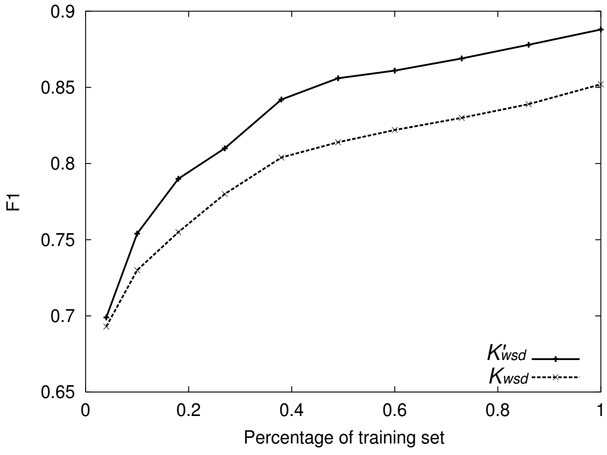

Figure 2: Learning curves for Catalan lexical sample task.

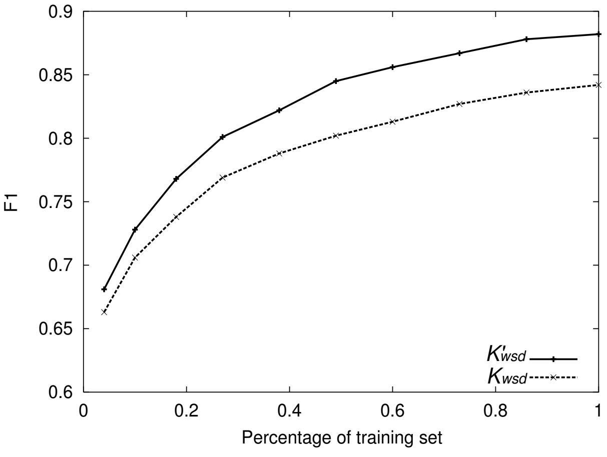

Figure 4: Learning curves for Spanish lexical sample task.

of our system outperforms the inter annotator agreement in both English and Spanish, achieving the upper bound performance.

We demonstrated that using external knowledge inside a supervised framework is a viable methodology to reduce the amount of training data required for learning. In our approach the external knowledge is represented by means of Domain Models automatically acquired from corpora in a totally unsupervised way. Experimental results show that the use of Domain Models allows us to reduce the amount of training data, opening an interesting research direction for all those NLP tasks for which the Knowledge Acquisition Bottleneck is a crucial problem. In particular we plan to apply the same methodology to Text Categorization, by exploiting the Domain Kernel to estimate the similarity among texts. In this implementation, our WSD system does not exploit syntactic information produced by a parser. For the future we plan to integrate such information by adding a tree kernel (i.e. a kernel function that evaluates the similarity among parse trees) to the kernel combination schema presented in this paper. Last but not least, we are going to apply our approach to develop supervised systems for all-words tasks, where the quantity of data available to train each word expert classifier is very low.

Acknowledgments

Alfio Gliozzo and Carlo Strapparava were partially supported by the EU project Meaning (IST-2001-34460). Claudio Giuliano was supported by the EU project Dot.Kom (IST-2001-34038). We would like to thank Oier Lopez de Lacalle for useful comments.

References

N. Cristianini and J. Shawe-Taylor. 2000. *An introduction to Support Vector Machines*. Cambridge University Press.

B. Decadt, V. Hoste, W. Daelemens, and A. van den Bosh. 2004. Gambl, genetic algorithm optimization of memory-based wsd. In *Proc. of Senseval-3*, Barcelona, July.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*.

A. Gliozzo, C. Strapparava, and I. Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18(3):275–299.

B. Magnini and G. Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000*, pages 1413–1418, Athens, Greece, June.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.

R. Mihalcea and P. Edmonds, editors. 2004. *Proceedings of SENSEVAL-3*, Barcelona, Spain, July.

R. Mihalcea and E. Faruque. 2004. Senselearner: Minimally supervised WSD for all words in open text. In *Proceedings of SENSEVAL-3*, Barcelona, Spain, July.

G. Salton and M.H. McGill. 1983. *Introduction to modern information retrieval*. McGraw-Hill, New York.

J. Shawe-Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

S. Small. 1980. *Word Expert Parsing: A Theory of Distributed Word-based Natural Language Understanding*. Ph.D. Thesis, Department of Computer Science, University of Maryland.

C. Strapparava, A. Gliozzo, and C. Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In *Proc. of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, Barcelona, Spain, July.

S.K.M. Wong, W. Ziarko, and P.C.N. Wong. 1985. Generalized vector space model in information retrieval. In *Proceedings of the 8^{th} ACM SIGIR Conference*.

D. Yarowsky and R. Florian. 2002. Evaluating sense disambiguation across diverse parameter space. *Natural Language Engineering*, 8(4):293–310.