Kernel Methods for Minimally Supervised WSD

Claudio Giuliano*
Fondazione Bruno Kessler – IRST

Alfio Massimiliano Gliozzo**
Fondazione Bruno Kessler – IRST

Carlo Strapparava†
Fondazione Bruno Kessler – IRST

We present a semi-supervised technique for word sense disambiguation that exploits external knowledge acquired in an unsupervised manner. In particular, we use a combination of basic kernel functions to independently estimate syntagmatic and domain similarity, building a set of word-expert classifiers that share a common domain model acquired from a large corpus of unlabeled data. The results show that the proposed approach achieves state-of-the-art performance on a wide range of lexical sample tasks and on the English all-words task of Senseval-3, although it uses a considerably smaller number of training examples than other methods.

1. Introduction

A significant challenge in many natural language processing tasks is to reduce the need for labeled training data while maintaining an acceptable performance. This is especially true for word sense disambiguation (WSD) because when moving from the somewhat artificial lexical-sample task to the more realistic all-words task it is practically impossible to collect a large number of training examples for each word sense. Thus, many supervised approaches, explicitly designed for the lexical-sample task, cannot be applied to the all-words task, even though they exhibit excellent performance. This has led to the somewhat paradoxical situation in which completely different methods have been developed for the two tasks, although they represent two sides of the same coin.

To address this problem, in recent work we presented a semi-supervised approach based on kernel methods for WSD (Strapparava, Gliozzo, and Giuliano 2004; Gliozzo, Giuliano, and Strapparava 2005; Giuliano, Gliozzo, and Strapparava 2006). In particular, we explored the following research directions: (1) independently modeling domain and syntagmatic aspects of sense distinction to improve feature representativeness; and (2) exploiting external knowledge acquired from unlabeled data, with the purpose of drastically reducing the amount of labeled training data. The first direction is based on the linguistic assumption that syntagmatic and domain (associative) relations are crucial for representing sense distinctions, but they are originated by different phenomena. Regarding the second direction, one can hope to obtain a more accurate prediction

* FBK-irst, via Sommarive 18, I-38050 Povo, Trento, Italy. E-mail: giuliano@fbk.eu.
** FBK-irst, via Sommarive 18, I-38050 Povo, Trento, Italy. E-mail: gliozzo@fbk.eu.
† FBK-irst, via Sommarive 18, I-38050 Povo, Trento, Italy. E-mail: strappa@fbk.eu.
Submission received: 23 December 2006; revised submission received: 28 February 2008; accepted for publication: 17 April 2008.

by taking into account unlabeled data relevant to the learning problem (Chapelle, Schölkopf, and Zien 2006). As a matter of fact, to test this hypothesis, most of the lexical sample tasks of Senseval-3 (Mihalcea and Edmonds 2004) were provided with a large amount of unlabeled training data, as well as the usual labeled training data. However, at that time, we were the only team to use the unlabeled data (Strapparava, Gliozzo, and Giuliano 2004).

In this article, we review our technique that combines domain and syntagmatic information in order to define a complete kernel for WSD. The rest of the article is organized as follows. In Section 2, we provide a general introduction to the kernel methods, in which we give the basis for understanding our approach. Exploiting kernel methods, we can define and combine individual kernels representing information from different sources in a principled way. After this introductory section, in Section 3 we present the kernels that we developed for WSD. This includes a detailed description of the individual kernels and the way we define the composite ones. We present our experiments in Section 4. The results obtained on a range of lexical-sample tasks and on the English all-words task of Senseval-3 (Mihalcea and Edmonds 2004) show that our approach achieves state-of-the-art performance. Finally, in Section 5, we offer conclusions and some directions for future research.

2. Kernel Methods

Kernel methods are a popular machine learning approach within the natural language processing community. They are theoretically well founded in statistical learning theory and have shown good empirical results in many applications (Vapnik 1999; Cristianini and Shawe-Taylor 2000; Schölkopf and Smola 2002; Shawe-Taylor and Cristianini 2004).

The strategy adopted by kernel methods consists of splitting the learning problem into two parts. They first embed the input data in a suitable feature space, and then use a linear algorithm to discover nonlinear patterns in the input space. Typically, the mapping is performed implicitly by a so-called **kernel function**. The kernel function is a similarity measure between the input data that depends exclusively on the specific data type and domain. A typical similarity function is the inner product between feature vectors. Characterizing the similarity of the inputs plays a crucial role in determining the success or failure of the learning algorithm, and it is one of the central questions in the field of machine learning.

Formally, the kernel is a function $K : X \times X \rightarrow \mathbb{R}$ that takes as input two data objects (e.g., vectors, texts, or parse trees) and outputs a real number characterizing their similarity, with the property that the function is symmetric and positive semi-definite. That is, for all $x_i, x_j \in X$ satisfies

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{1}$$

where ϕ is an (implicit) mapping from X to an (inner product) feature space $\mathcal{F}$.

Kernels are used inside learning algorithms such as support vector machines (SVM) or kernel perceptrons as the interface between the algorithm and the data. The kernel function is then the only domain specific element of the system, while the learning algorithm is a general purpose component.

The idea behind the SVM (one of the best known kernel-based learning algorithms) is to map the set of training data into a high-dimensional feature space $\mathcal{F}$ via a mapping function $\phi : X \rightarrow \mathcal{F}$, and construct a separating hyperplane with maximum margin (i.e.,

the minimum distance between the hyperplane and data points) in that space. The use of an appropriate non-linear transformation ϕ of the input yields a nonlinear decision boundary in the input space. Kernel functions make possible the use of feature spaces with an exponential or even infinite number of dimensions. Instead of performing the explicit feature mapping ϕ, one can use a kernel function, which permits the (efficient) computation of inner products in high-dimensional feature spaces without explicitly carrying out the mapping ϕ. This is called the **kernel trick** in the machine learning literature (Boser, Guyon, and Vapnik 1992).

Finally, we point out the theoretical tools required to create new kernels, and combine individual kernels to form composite ones. Of course, not every similarity function is a valid kernel because, by definition, kernels should be equivalent to some inner product in a feature space. The function $K : X \times X \rightarrow \mathbb{R}$ is a valid kernel provided that its kernel matrices[1] are positive semi-definite[2] for all training sets $S = \{x_1, ..., x_l\}$, the so-called finitely positive semi-definite property. Note that defining similarity measures by means of kernels may be more intuitive than performing the explicit mapping in the feature space. Furthermore, this formulation does not require the set X to be a vector space: for example, we shall define kernels that take strings as input.

This result is not only useful because it opens new perspectives to define kernel functions that only implicitly correspond to a feature mapping ϕ. Another consequence is that it can be used to prove a set of rules for combining basic kernels to obtain composite ones. This will allow us to integrate heterogeneous sources of information in a simple and effective way. We shall use the following properties of kernels to define our composite kernels. Let k_1 and k_2 be kernels over $X \times X$; then the following functions are kernels:

- $k(x_i, x_j) = k_1(x_i, x_j) + k_2(x_i, x_j)$
- $k(x_i, x_j) = c \cdot k_1(x_i, x_j), c \in \mathbb{R}^+$
- $k(x_i, x_j) = \frac{k_1(x_i, x_j)}{\sqrt{k_1(x_i, x_i) \times k_1(x_j, x_j)}}$ (normalization)

In summary, we can define a kernel function by following different strategies: (1) providing an explicit feature mapping $\phi : X \rightarrow \mathbb{R}^n$; (2) defining a similarity function that is symmetric and positive semi-definite; and (3) composing different valid kernels, using the closure properties of kernels. This forms the basis for the approach described in the following section.

3. Kernel Methods for WSD

Our approach to WSD consists of representing linguistic phenomena independently and then defining a combination method to integrate them. As described in the previous section, the kernel function is the only task-specific component of the learning algorithm. Thus, to develop a WSD system, we only need to define appropriate kernel functions to represent the domain and syntagmatic aspects of sense distinction and, second, exploit the properties of kernel functions to define a composite kernel to combine and extend the individual kernels.

The resulting WSD system consists of two families of kernels: the domain and the syntagmatic kernels. The former family, described in Section 3.1, models the domain

1 Given a set of vectors $S = \{x_1, ..., x_l\}$, the kernel matrix $\mathbf{K}$ is defined as the $l \times l$ matrix $\mathbf{K}$ whose entries are $\mathbf{K}_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where k is a kernel function that evaluates the inner products in a feature space with feature map ϕ.

2 A symmetric matrix is positive semi-definite if its eigenvalues are all non-negative. Actually, as we will see in Section 3.2 using Proposition 1, it is quite easy to verify this property.

Table 1
An example of a domain matrix.

	Medicine	Computer Science
HIV	1	0
AIDS	1	0
virus	0.5	0.5
laptop	0	1

aspects of sense distinction; it is composed of the domain kernel (K_D) and the bag-of-words kernel (K_{BoW}). The latter, described in Section 3.2, represents the syntagmatic aspects of sense distinction; it is composed of the collocation kernel (K_{Coll}) and the part-of-speech kernel (K_{PoS}). Finally, Section 3.3 describes the composite kernel for WSD.

3.1 Domain Kernels

It has been shown that domain information is fundamental for WSD (Magnini et al. 2002). For instance, the (domain) polysemy between the computer science and the medicine senses of the word *virus* can be solved by considering the domain of the context in which it appears. Gliozzo, Strapparava, and Dagan (2004) proposed a WSD method that exploits only domain information.

In the context of kernel methods, domain information can be exploited by defining a kernel function that estimates the domain similarity between the contexts of the words to be disambiguated. The simplest method to estimate the domain similarity between two texts is to compute the cosine similarity of their vector representations in the vector space model (VSM). The VSM is a k-dimensional space $\mathbb{R}^k$, in which the text t_j is represented by a vector $\vec{t_j}$, where the i^{th} component is the term frequency of the term w_i in t_j. However, such an approach does not deal well with lexical variability and ambiguity. For instance, despite the fact that the sentences *He is affected by AIDS* and *HIV is a virus* express closely-related concepts, their similarity is zero in the VSM because they have no words in common (they are represented by orthogonal vectors). On the other hand, due to the ambiguity of the word *virus*, the similarity between the sentences *The laptop has been infected by a virus* and *HIV is a virus* is greater than zero, even though they convey very different messages.

To overcome this problem, we introduce the domain model (DM) and show how to use it to define a domain VSM in which texts and terms are represented in a uniform way. A DM is composed of soft clusters of terms. Each cluster represents a semantic domain, that is, 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.

The matrix $\mathbf{D}$ is used to define a function $\mathcal{D} : \mathbb{R}^k \rightarrow \mathbb{R}^{k'}$, that maps the vector $\vec{t_j}$ represented in the standard VSM into the vector $\vec{t_j'}$ in the domain VSM. $\mathcal{D}$ is defined as follows:[3]

$$\mathcal{D}(\vec{t_j}) = \vec{t_j}(\mathbf{I^{IDF}D}) = \vec{t_j'} \tag{2}$$

3 In Wong, Ziarko, and Wong (1985), Equation (2) is used to define a generalized vector space model, of which the domain VSM is a particular instance.

where $\vec{t_j}$ is represented as a row vector, $\mathbf{I^{IDF}}$ is a $k \times k$ diagonal matrix such that $i_{i,i}^{IDF} = IDF(w_i)$, and $IDF(w_i)$ is the inverse document frequency of w_i.

In the domain space, the similarity 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 strongly associated with the medicine domain.

A DM can be estimated from manually constructed lexical resources, such as WordNet Domains (Magnini and Cavaglià 2000), or by performing a term-clustering process on a (large) corpus. However, the second approach is more attractive because it allows us to automatically acquire DMs for different languages and domains.

In Gliozzo, Giuliano, and Strapparava (2005), we use singular valued decomposition (SVD) to acquire DMs from a corpus represented by its term-by-document matrix $\mathbf{T}$, in a unsupervised way.[4] SVD decomposes the term-by-document matrix $\mathbf{T}$ into three matrixes $\mathbf{T} \simeq \mathbf{V}\mathbf{\Sigma}_{\mathbf{k}'}\mathbf{U}^T$, where $\mathbf{V}$ and $\mathbf{U}$ are orthogonal matrices (i.e., $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$) whose columns are the eigenvectors of $\mathbf{T}\mathbf{T}^T$ and $\mathbf{T}^T\mathbf{T}$, respectively, and $\mathbf{\Sigma}_{\mathbf{k}'}$ 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. Under this setting, we define the domain matrix $\mathbf{D}$ as follows:

$$\mathbf{D} = \mathbf{I^N}\mathbf{V}\sqrt{\mathbf{\Sigma}_{\mathbf{k}'}} \tag{3}$$

where $\mathbf{I^N}$ is a diagonal matrix such that $\mathbf{i_{i,i}^N} = \frac{1}{\sqrt{\langle \vec{w_i'}, \vec{w_i'} \rangle}}$, $\vec{w_i'}$ is the i^{th} row of the matrix $\mathbf{V}\sqrt{\mathbf{\Sigma}_{\mathbf{k}'}}$.[5]

Note that in this case, with respect to Table 1, the domains are represented by the columns of the matrix $\mathbf{D}$ and they do not have an explicit name. By using a small number of domains, we can define a very compact representation of the DM and, consequently, reduce the memory requirements while preserving most of the information. There exist very efficient algorithms to perform the SVD process on sparse matrices, allowing us to perform this operation on large corpora in a very limited time and with reduced memory requirements.[6]

Therefore, we can define the domain kernel to estimate the domain similarity between the contexts of the words to be disambiguated. It is a variant of the latent semantic kernel (Shawe-Taylor and Cristianini 2004), in which a DM 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 explicitly defined as follows:

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

where $\mathcal{D}$ is the domain mapping defined in Equation (2).

4 The SVD algorithm was first adopted to perform latent semantic analysis of terms and latent semantic indexing of documents in large corpora (Deerwester et al. 1990).

5 When $\mathbf{D}$ is substituted in Equation (2) the domain VSM 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 used in information retrieval.

6 To perform the SVD, we used LIBSVDC, an optimized package for sparse matrices that allows us to perform this step in a few minutes even for large corpora. It can be downloaded from `http://tedlab.mit.edu/~dr/SVDLIBC/`.

A standard approach for detecting topic (domain) similarity is to extract bag-of-words features from a wide window of text around the words to be disambiguated. Based on this representation, we define a linear kernel called the **bag-of-words** kernel (K_{BoW}). K_{BoW} is a particular case of the domain kernel in which $\mathbf{D} = \mathbf{I}$ in Equation (2), where $\mathbf{I}$ is the identity matrix. The BoW kernel does not require a DM; therefore, it can be applied to the strictly supervised settings, in which external knowledge is not available.

To summarize, the domain kernel allows us to plug external knowledge into the supervised learning process; it will be compared and combined with the standard bag-of-words approach in Section 4. In the following section, we shall see that domain models are also useful for defining soft-matching collocation kernels.

3.2 Syntagmatic Kernels

Collocations (such as bigrams and trigrams) extracted from the local context of the word to be disambiguated are typically used to capture syntagmatic relations (Yarowsky 1994). However, traditional approaches to WSD fail to represent non-contiguous or shifted collocations, and fail to consider lexical variability. For example, suppose we have to disambiguate the verb *to score* in the sentence *Ronaldo scored the first goal*, given the labeled example *The football player scored two goals in the second half* as training. A traditional approach has no clues to return the right answer because the two sentences have no features in common.

The use of kernels on strings allows us to overcome the aforementioned problems by representing (non-contiguous) collocations and exploiting external lexical knowledge sources to define non-zero measures of similarity between words (soft-matching criteria). In this formulation, words taken in their context are compared by kernels that sum the number of common (non-contiguous) collocations of words, considering lexical variability, and part-of-speech tags, avoiding an explicit feature mapping that would lead to an exponential number of features.

String kernels (or sequence kernels) are a family of kernel functions developed to compute the inner product among images of strings in high-dimensional feature space using dynamic programming techniques. The gap-weighted subsequences kernel is one of the most general types of kernel based on sequences. Roughly speaking, it compares two strings by means of the number of contiguous and non-contiguous substrings of a given length they have in common. Non-contiguous occurrences are penalized according to the number of gaps they contain. Formally, let Σ be an alphabet of $|\Sigma|$ symbols, and $s = s_1 s_2 \dots s_{|s|}$ be a finite sequence over Σ (i.e., $s_i \in \Sigma, 1 \leqslant i \leqslant |s|$). Let $\mathbf{i} = [i_1, i_2, \dots, i_n]$, with $1 \leqslant i_1 < i_2 < \dots < i_n \leqslant |s|$, be a subset of the indices in s; we will denote as $s[\mathbf{i}] \in \Sigma^n$ the subsequence $s_{i_1} s_{i_2} \dots s_{i_n}$. Note that $s[\mathbf{i}]$ does not necessarily form a contiguous subsequence of s; for example, if s is the sequence "Ronaldo scored the first goal" and $\mathbf{i} = [2, 5]$, then $s[\mathbf{i}]$ is "scored goal". The length spanned by $s[\mathbf{i}]$ in s is $l(\mathbf{i}) = i_n - i_1 + 1$. The feature space associated with the gap-weighted subsequences kernel of length n is indexed by $I = \Sigma^n$, with the embedding given by

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

where $0 < \lambda \leqslant 1$ is the decay factor used to penalize non-contiguous subsequences.[7] The associate kernel is defined as

$$K_n(s,t) = \langle \phi^n(s), \phi^n(t) \rangle = \sum_{u \in \Sigma^n} \phi_u^n(s) \phi_u^n(t) \tag{6}$$

An explicit computation of Equation (6) is unfeasible even for small values of n. To evaluate K_n more efficiently, we use the recursive formulation based on a dynamic programming implementation (Lodhi et al. 2002; Saunders, Tschach, and Shawe-Taylor 2002; Cancedda et al. 2003). It is defined in the following equations:

$$K'_0(s,t) = 1, \forall s,t \tag{7}$$

$$K'_i(s,t) = 0, \text{if } \min(|s|,|t|) < i \tag{8}$$

$$K''_i(s,t) = 0, \text{if } \min(|s|,|t|) < i \tag{9}$$

$$K''_i(sx,ty) = \begin{cases} \lambda K''_i(sx,t) & \text{if } x \neq y; \\ \lambda K''_i(sx,t) + \lambda^2 K'_{i-1}(s,t) & \text{otherwise.} \end{cases} \tag{10}$$

$$K'_i(sx,t) = \lambda K'_i(s,t) + K''_i(sx,t) \tag{11}$$

$$K_n(s,t) = 0, \text{if } \min(|s|,|t|) < n \tag{12}$$

$$K_n(sx,t) = K_n(s,t) + \sum_{j:t_j=x} \lambda^2 K'_{n-1}(s, t[1:j-1]) \tag{13}$$

where K'_n and K''_n are auxiliary functions with a similar definition to K_n used to facilitate the computation. Based on these definitions, K_n can be computed in $O(n|s||t|)$. Using this recursive definition, it turns out that computing all kernel values for subsequences of lengths up to n is not significantly more costly than computing the kernel for n only.

The syntagmatic kernel is defined as a sum of gap-weighted subsequences kernels that operate at word and part-of-speech tag level. In particular, following the approach proposed by Cancedda et al. (2003), it is possible to adapt sequence kernels to operate at word level by instancing the alphabet Σ with the vocabulary $\mathcal{V} = \{w_1, w_2, \ldots, w_k\}$. Moreover, we restrict the generic definition of the gap-weighted subsequences kernel to recognize collocations in the local context of a specified word. The resulting kernel, called the **n-gram** collocation kernel (K^n_{Coll}), operates on sequences of lemmata around a specified word l_0 (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. Analogously, we define the part-of-speech kernel (K^n_{PoS}) to operate on sequences of part-of-speech tags $p_{-3}, p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}$, where p_0 is the part-of-speech tag of l_0.

The collocation kernel and the part-of-speech kernel are defined by Equations (14) and (15), respectively.

$$K_{Coll}(s,t) = \sum_{l=1}^{n} K^l_{Coll}(s,t) \tag{14}$$

7 Notice that by choosing $\lambda = 1$, sparse subsequences are not penalized. On the other hand, the kernel does not take into account sparse subsequences with $\lambda \to 0$.

$$K_{PoS}(s,t) = \sum_{l=1}^{n} K^{l}_{PoS}(s,t) \tag{15}$$

Both kernels depend on the parameter n, the length of the non-contiguous subsequences, and λ, the decay factor. For example, K^{2}_{Coll} allows us to represent all (sparse) bigrams in the local context of a word. Finally, the syntagmatic kernel is defined as

$$K_{Synt}(s,t) = K_{Coll}(s,t) + K_{PoS}(s,t) \tag{16}$$

In the preceding definition, only exact word-matches contribute to the similarity. To solve this problem, external lexical knowledge is fed into the supervised learning process, allowing us to define the **soft-matching** collocation kernel. In particular, we define two alternative soft-matching criteria by exploiting synonymy relations in WordNet and DMs acquired from corpora. Both criteria are based on the assumption that every word in a sentence can be substituted by another preserving the original meaning, if these words are paradigmatically related (e.g., synonyms, hyponyms, or domain related words). For example, if we consider as equivalent the terms *Ronaldo* and *football player*, then the sentence *The football player scores the first goal* is equivalent to *Ronaldo scores the first goal*, providing a strong evidence to disambiguate the verb *to score* in the second sentence.

Following the approach proposed by Shawe-Taylor and Cristianini (2004), the soft-matching gap-weighted subsequences kernel is now calculated recursively using Equations (7)–(9), (11), and (12), replacing Equation (10) by the equation:

$$K''_i(sx,ty) = \lambda K''_i(sx,t) + \lambda^2 a_{xy} K'_{i-1}(s,t), \forall x,y \tag{17}$$

and modifying Equation (13) to:

$$K_n(sx,t) = K_n(s,t) + \sum_{j}^{|t|} \lambda^2 a_{xt_j} K'_{n-1}(s, t[1:j-1]) \tag{18}$$

where a_{xy} are entries in a similarity matrix $\mathbf{A}$ between terms. In order to ensure that the resulting kernel is still valid, $\mathbf{A}$ must be positive semi-definite.

In the following sections, we describe the two alternative soft-matching criteria based on WordNet Synonymy and Domain Proximity, respectively. To show that the similarity matrices are positive semi-definite, we use the following result.

Proposition 1
A matrix $\mathbf{A}$ is positive semi-definite if and only if $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ for some real matrix $\mathbf{B}$.
The proof is given in Shawe-Taylor and Cristianini (2004).

WordNet Synonymy. The first soft-matching criterion is based on WordNet[8] to define a similarity matrix between words. In particular, we substitute two words if they are synonyms. To this end, a word is represented as vector whose dimensions are associated

8 We used WordNet 1.7.1 and MultiWordNet for English and Italian experiments, respectively.

with the synsets. Formally, we define the term-by-synset matrix $\mathbf{S}$ as the matrix whose rows are indexed by the terms and whose columns are indexed by the synsets. The (i,j)th entry of $\mathbf{S}$ is 1 if the synset s_j contains the term w_i; 0 otherwise. The matrix $\mathbf{S}$ gives rise to the similarity matrix $\mathbf{A} = \mathbf{S}\mathbf{S}^T$ between terms. Because $\mathbf{A}$ can be rewritten as $\mathbf{A} = (\mathbf{S}^T)^T\mathbf{S}^T = \mathbf{B}^T\mathbf{B}$, it follows directly from Proposition 1 that it is positive semi-definite.

Domain Proximity. The second soft-matching criterion exploits the domain models introduced in Section 3.1 to define a similarity matrix between words. Once a DM has been defined by the matrix $\mathbf{D}$, the domain space is a k' dimensional space, in which both texts and terms are represented by means of domain vectors, that is, vectors representing the domain relevances among the linguistic object and each domain. The domain vector $\vec{w}'_i$ for the term $w_i \in \mathcal{V}$ is the i^{th} row of $\mathbf{D}$, where $\mathcal{V} = \{w_1, w_2, \ldots, w_k\}$ is the vocabulary of the corpus. The term-by-domain matrix $\mathbf{D}$ gives rise to the similarity matrix $\mathbf{A} = \mathbf{D}\mathbf{D}^T$ between terms. It follows by Proposition 1 that $\mathbf{A}$ is positive semi-definite.

We shall show that the syntagmatic kernel is more effective than standard bigrams and trigrams of lemmata and part-of-speech tags typically used as features in WSD.

3.3 Composite Kernel

Having defined all the individual kernels representing syntagmatic and domain aspects of sense distinction, we can define the composite kernel to combine and extend the individual kernels. The closure properties of the kernel functions allows us to define the composite kernel as

$$K_C(x_i, x_j) = \sum_{l=1}^{n} \frac{K_l(x_i, x_j)}{\sqrt{K_l(x_j, x_j)K_l(x_i, x_i)}} \tag{19}$$

where K_l is a valid individual kernel. The individual kernels are normalized—this plays an important role in allowing us to integrate information from heterogeneous feature spaces.

Recent work (Moschitti 2004; Gliozzo, Giuliano, and Strapparava 2005; Zhao and Grishman 2005; Giuliano, Lavelli, and Romano 2006) has empirically shown the effectiveness of combining kernels in this way: The composite kernel consistently improves the performance of the individual ones. In addition, this formulation allows us to evaluate the individual contribution of each information source.

In order to show the effectiveness of the proposed domain model in supervised learning, we defined two WSD kernels, K_{wsd} and K'_{wsd}. They are completely specified by the n individual kernels that compose them in Equation (19).

K_{wsd} is composed by K_{Coll}, K_{PoS}, and K_{BoW};
K'_{wsd} is composed by K_{Coll}, K_{PoS}, K_{BoW}, and K_D.

The only difference between the two is that K'_{wsd} uses the domain kernel K_D to exploit external knowledge while K_{wsd} only uses the labeled training data.

4. Evaluation

Experiments were carried out on various tasks of Senseval-3 (Mihalcea and Edmonds 2004). First of all, we conducted a preliminary set of experiments on the Catalan, English, Italian, and Spanish lexical-sample tasks; the results are shown in Section 4.1. Second, in order to show the general applicability of the proposed method, we evaluated the system on the English all-words task; the results are presented in Section 4.2.

All the experiments were performed using the SVM package (Chang and Lin 2001) customized to embed our own kernels. The parameters were optimized by five-fold cross-validation on the training set.

4.1 Lexical-Sample Tasks

In this section, we report the evaluation of our method on the Catalan, English, Italian, and Spanish lexical-sample tasks of Senseval-3 (Mihalcea and Edmonds 2004). Table 2 describes the tasks we have considered. For each task, it summarizes the number of words to be disambiguated, the mean polysemy, the size of the labeled training set, the size of the test set, and the size of the unlabeled training set, respectively. For the Catalan, Italian, and Spanish tasks, we acquired the DMs from the unlabeled corpora made available by the task organizers. For the English task, we used a DM acquired from the British National Corpus (BNC) as the task organizers have not provided any unlabeled training data. The objectives of these experiments are to (a) estimate the impact of different knowledge sources in WSD; (b) study the effectiveness of the kernel combination; (c) understand the benefits of plugging external information in a supervised framework; and (d) verify the portability of our methodology to different languages.

4.1.1 Results. Table 3 reports the results of the individual kernels K_{BoW}, K_D, K_{Coll}, and K_{PoS} and their combinations K_{wsd} and K'_{wsd} (the baselines for the tasks are reported in Table 5). In our experiments, the parameters n and λ (see Equation (5)) are optimized by five-fold cross-validation. For K^n_{Coll}, we obtained the best results with $n = 2$ and $\lambda = 0.5$. For K^n_{PoS}, $n = 3$ and $\lambda \to 0$. The domain cardinality k' was set to 50. Table 4 shows the performance of the syntagmatic kernel in different configurations: hard and soft matching. As a baseline, we report the result of a standard approach consisting of explicit bigrams and trigrams of words and part-of-speech tags around the words to be disambiguated (Yarowsky 1994). We evaluated the impact of the domain kernel on the overall performance by comparing the learning curves of K'_{wsd} and K_{wsd} on the four lexical-sample tasks. Figure 1 shows the results of our experiments. The points of the learning curves are obtained by sampling the same percentage of training examples for

Table 2
Description of the lexical-sample tasks of Senseval-3.

Task	*#w*	*mean polysemy*	*#train*	*#test*	*#unlab*
Catalan	27	3.11	4,469	2,253	23,935
English	57	6.47	7,860	3,944	-
Italian	45	6.30	5,145	2,439	74,788
Spanish	46	3.30	8,430	4,195	61,252

Table 3
The performance (F1) of the basic and composite kernels on the Catalan, English, Italian, and Spanish lexical-sample tasks of Semeval-3.

Kernel	*Catalan*	*English*	*Italian*	*Spanish*
K_{BoW}	81.3	63.7	43.3	78.2
K_D	85.2	65.5	44.5	84.4
K_{Coll}	84.2	68.5	54.0	83.6
K_{PoS}	79.6	64.0	44.4	79.5
K_{wsd}	85.2	69.7	53.1	84.2
K'_{wsd}	89.0	73.3	61.3	88.2

Table 4
Performance (F1) of the syntagmatic kernel for the Catalan, English, Italian, and Spanish lexical-sample tasks of Semeval-3.

Method	*Catalan*	*English*	*Italian*	*Spanish*
Bigrams and trigrams	82.6	67.3	51.0	81.9
Hard matching	83.8	67.7	51.9	82.9
Soft matching (WordNet)	-	67.3	51.3	-
Soft matching (Domain proximity)	84.2	68.5	54.0	83.6

each word. Finally, Table 5 summarizes the results we obtained, providing a comparison with the state of the art.

4.1.2 Discussion. Table 3 shows that domain information and syntagmatic information are crucial for WSD, and their combination significantly outperforms the individual kernels, showing the effectiveness of the kernel combination method.

In addition, the domain kernel K_D outperforms the bag-of-words kernel K_{BoW}, and the composite kernel K'_{wsd} that makes use of domain information outperforms the one K_{wsd} based only on the labeled training data, demonstrating our assumption (see Section 3).

Table 4 shows that the syntagmatic kernel outperforms the baseline (bigrams and trigrams) in any configuration (hard-/soft-matching). The soft-matching criteria further improve the classification performance. It is interesting to note that the domain proximity obtained better results than WordNet synonymy (note that we do not have a Catalan or a Spanish WordNet). The different results observed for Italian and English using the domain proximity soft-matching criterion are probably due to the small size of the unlabeled English corpus.

Figure 1 shows that K'_{wsd} outperforms K_{wsd} on all lexical sample tasks, even with a small number of examples. It is worth noting, as reported in Table 5, that K'_{wsd} achieves the same performance as K_{wsd} using about half of the labeled training data. This result shows that the proposed semi-supervised learning approach consisting of acquiring domain models from unlabeled corpora is effective, as it allows us to drastically reduce the amount of labeled training data and provide a viable solution for the knowledge acquisition bottleneck problem in WSD.

To the best of our knowledge, K'_{wsd} turns out to be the best system for all the tested tasks of Senseval-3, further improving the state of the art by 0.4% to 8.2% for English and Italian, respectively. Finally, we have demonstrated the language independency

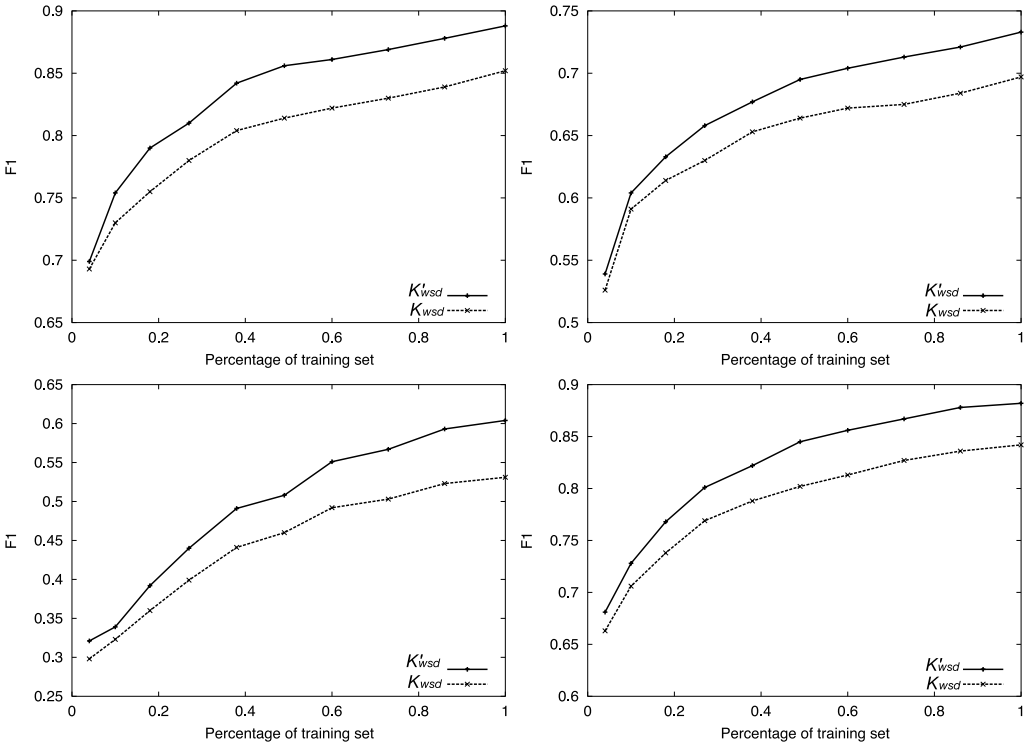

Figure 1
From left to right, top to bottom, learning curves for the Catalan, English, Italian, and Spanish lexical-sample tasks of Semeval-3.

of our approach. The DMs have been acquired for different languages from different unlabeled corpora by adopting exactly the same methodology, without requiring any external lexical resource or ad hoc rule.

4.2 All-Words Task

Encouraged by the excellent results obtained on the lexical-sample tasks, we evaluated our approach on the all-words task, in which a very small amount of labeled training

Table 5
Comparative evaluation on the lexical sample tasks.

Task	*MF*	*Agreement*	*BEST*	K_{wsd}	K'_{wsd}	*DM+*	*% of training*
Catalan	66.3	93.1	85.2	85.2	**89.0**	3.8	46
English	55.2	67.3	72.9	69.7	**73.3**	3.6	54
Italian	18.0	89.0	53.1	53.1	**61.3**	8.2	51
Spanish	67.7	85.3	84.2	84.2	**88.2**	4.0	50

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., $K'_{wsd} - K_{wsd}$), and the percentage of sense-tagged examples required by K'_{wsd} to achieve the same performance as K_{wsd} with full training.

Table 6
The performance (F1) of the basic kernels and composite kernels on the English all-words task of Senseval-3.

	basic kernels					*composite kernels*		
	K_D^{bnc}	K_D^{sem}	K_{BoW}	K_{PoS}	K_{Coll}	K_{wsd}	$\hat{K}_{wsd}^{bnc}$	$\hat{K}_{wsd}^{sem}$
F1	63.0	63.2	63.2	63.4	64.0	64.4	65.0	**65.2**

data is typically available. We performed the evaluation on the English all-words task of Senseval-3 (Snyder and Palmer 2004). The test set was extracted from two *Wall Street Journal* articles and one text from the Brown Corpus. The test set consists of 945 words (2,041 word occurrences) to be disambiguated with WordNet 1.7.1 senses. The inter-annotator agreement rate in the preparation of the corpus was approximately 72.5%. The most frequent (MF) baseline using the first WordNet sense heuristic obtained 60.9%.

We have trained and tested the system exploiting the following resources: (1) WordNet 1.7.1 as sense repository; (2) SemCor,[9] considering only those words appearing in the Senseval-3 all-words data set—we extracted about 61,700 tagged examples that constitute the only labeled training set exploited by the system; and (3) the BNC, from which we extracted the unlabeled training data.

4.2.1 Results. We trained 734 word-expert classifiers on the SemCor corpus. The labeled examples for each classifier range from a minimum of one example to a maximum of 2,275 examples. We return a random sense for those words that have no training examples in SemCor.[10] We have acquired two DMs, one from the BNC (i.e., $\hat{K}_D^{bnc}$; the same we used in the lexical-sample task) and one from SemCor (i.e., $\hat{K}_D^{sem}$), obtaining a slightly better performance with the latter.

Table 6 shows the performance of the individual kernels K_{BoW}, K_D, K_{Coll}, and K_{PoS}, and their composite kernels K_{wsd}, $\hat{K}_D^{bnc}$ and $\hat{K}_D^{sem}$.

Since for 210 words in the test set we have no training examples, to better understand the results obtained, we performed an evaluation on the subset of the test set for which at least one training example is available in SemCor. Evaluating only on these words the performance increases from 65.2% to 70.0%, and the most frequent baseline becomes 65.7%. Tables 7 and 8 present a more detailed analysis that considers results grouped according to the amount of training available and the mean polysemy of the words in the test set, excluding from the data set the monosemous words. Table 7 shows the results (F1) of $\hat{K}_{wsd}^{sem}$ at different ranges of polysemy. Table 8 presents the results (F1) of $\hat{K}_{wsd}^{sem}$ on those words that have a given number of training examples. This evaluation is limited to the best composite kernel $\hat{K}_{wsd}^{sem}$.

4.2.2 Discussion. We compared our approach with the three best systems that participated in the English all-words task of Senseval-3. The best system (Decadt et al. 2004) has comparable performance (65.2) to ours; however, it uses a larger training set composed of 563,129 sense-tagged words. The training corpus was built by merging

9 Texts semantically annotated with WordNet 1.6 senses (created at Princeton University), and automatically mapped to WordNet 1.7, WordNet 1.7.1, and WordNet 2.0. Downloadable from `http://www.cs.unt.edu/~rada/downloads.html`.

10 Note that for these words the WordNet first sense is not necessarily the most frequent sense.

Table 7
The performance (F1) of $\hat{K}_{wsd}^{sem}$ at different ranges of polysemy. Most Frequent baseline (MF) is also reported.

	Range of polysemy						
	2–5	6–10	11–15	16–20	21–25	26–30	31+
$\hat{K}_{wsd}^{sem}$	73.2	61.4	59.1	33.8	55.2	50.2	37.3
MF	70.0	53.4	56.4	25.7	47.2	39.0	21.7

Table 8
The performance (F1) of $\hat{K}_{wsd}^{sem}$ on words with a given number of training examples. Most Frequent baseline (MF) and mean polysemy for each partition are also reported.

	Range of training examples				
	1–10	11–50	51–100	101–200	201+
$\hat{K}_{wsd}^{sem}$	76.1	70.8	54.2	67.4	60.0
MF	73.5	66.4	49.4	63.2	53.0
Mean polysemy	3	5	7	9	15

SemCor, and English lexical-sample and all-words data sets taken from all the previous editions of Senseval. The system proposed by Mihalcea and Faruque (2004) scored second (64.6). The dimension of their training set is comparable to ours; however, they also use additional information drawn from WordNet to derive semantic generalizations using syntactic dependencies. Finally, the third system (Yuret 2004) obtained 64.1 using a larger training data set (Semcor, DSO corpus of sense-tagged English, OpenMind Word Expert, Senseval-2, and Senseval-3 lexical-sample tasks).

The small difference between the two domain models seems to indicate that a limited amount of unlabeled data is sufficient to improve the overall performance, and the use of unlabeled data taken from the training set helps to slightly improve the overall performance. However, the domain model can be acquired from a different corpus (e.g., the BNC) without significantly affecting the overall performance.

Finally, the results reported in Tables 7 and 8 show that our approach is able to disambiguate with good accuracy ($F1 = 76\%$) words with a number of training examples that ranges from 1 to 10, outperforming the most frequent baseline by 3%. This is an interesting result given the extremely small number of training examples available. On the other hand, the more training is available for a given word, the more polysemous that word is. Nevertheless, the algorithm always outperforms the baseline and has a more significant difference for increasing values of the mean polysemy (from 3% to 16%). These results, together with the ones obtained in the lexical sample tasks, show that the domain kernel is able to boost the overall performance when little training data are available, as well as with enough training data. The benefit is even more pronounced for the latter case, even though the disambiguation task is more complex due to the high polysemy of highly frequent words.

5. Conclusions

This article summarizes the results of a word expert semi-supervised algorithm for WSD based on a combination of kernel functions. First, we evaluated our methodology

on four lexical-sample tasks of Senseval-3, significantly improving the state of the art for all of them. In particular, 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. Then, we applied the method so defined to the English all-words task of Senseval-3, achieving state-of-the-art performance while requiring less labeled training data compared to the other systems we have found in the literature.

Some slight improvement may be possible by exploiting syntactic information produced by a parser. In the framework of kernel methods, this expansion can be done by adding a tree kernel (i.e., a kernel function that evaluates the similarity among parse trees) to our composite kernel. However, the performance achieved is close to the upper bound, if we consider the inter-annotator agreement as an indication of the upper-bound performance.

Finally, we think that our semi-supervised approach is at the moment an effective solution for developing a sense-tagging system. Indeed, we tested the system on the English lexical-sample task of SemEval 2007, still obtaining state-of-the-art performance (Pradhan et al. 2007). Therefore, we plan to make available an optimized version of our system, and to exploit it for ontology learning, textual entailment, and information retrieval.

Acknowledgments

Claudio Giuliano was supported by the X-Media project (`www.x-media-project.org`), sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant IST-FP6-026978. Alfio Massimiliano Gliozzo and Carlo Strapparava were supported by the ONTOTEXT project, sponsored by the Autonomous Province of Trento under the FUP-2004 research program.

References

Boser, Bernhard, Isabelle Guyon, and Vladimir Vapnik. 1992. A training algorithm for optimal margin classifier. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA.

Cancedda, Nicola, Eric Gaussier, Cyril Goutte, and Jean-Michel Renders. 2003. Word-sequence kernels. *Journal of Machine Learning Research*, 32(6):1059–1082.

Chang, Chih-Chung and Chih-Jen Lin, 2001. *LIBSVM: A library for support vector machines*. Software available at `www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chapelle, Olivier, Bernhard Schölkopf, and Alexander Zien. 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

Cristianini, Nello and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press.

Decadt, Bart, Veronique Hoste, Walter Daelemans, and Antal van den Bosch. 2004. GAMBL, genetic algorithm optimization of memory-based WSD. In *Proceedings of Senseval-3*, pages 108–112, Barcelona.

Deerwester, Scott, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41:391–407.

Giuliano, Claudio, Alfio Gliozzo, and Carlo Strapparava. 2006. Syntagmatic kernels: A word sense disambiguation case study. In *In Proceedings of the EACL-06 Workshop on Learning Structured Information in Natural Language Applications*, pages 57–63, Trento.

Giuliano, Claudio, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 401–408, Trento.

Gliozzo, Alfio, Claudio Giuliano, and Carlo Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 403–410, Ann Arbor, MI.

Gliozzo, Alfio, Carlo Strapparava, and Ido Dagan. 2004. Unsupervised and supervised exploitation of semantic

domains in lexical disambiguation. *Computer Speech and Language*, 18(3):275–299.
Lodhi, Huma, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(3):419–444.
Magnini, Bernardo and Gabriela Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000*, pages 1413–1418, Athens.
Magnini, Bernardo, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.
Mihalcea, Rada and Phil Edmonds, editors. 2004. *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona.
Mihalcea, Rada and Ehsanul Faruque. 2004. SenseLearner: Minimally supervised WSD for all words in open text. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 155–158, Barcelona.
Moschitti, Alessandro. 2004. A study on convolution kernels for shallow statistic parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 335–342, Barcelona.
Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague.
Salton, Gerard and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.
Saunders, Craig, Hauke Tschach, and John Shawe-Taylor. 2002. Syllables and other string kernel extensions. In *Proceedings of 19th International Conference on Machine Learning (ICML02)*, pages 530–537, Sydney.
Schölkopf, Bernhard and Alexander Smola. 2002. *Learning with Kernels*. MIT Press, Cambridge, MA.
Shawe-Taylor, John and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.
Snyder, Benjamin and Martha Palmer. 2004. The English all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona.
Strapparava, Carlo, Alfio Gliozzo, and Claudio Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234, Barcelona.
Vapnik, Vladimir N. 1999. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, Berlin.
Wong, S. K. M., Wojciech Ziarko, and Patrick C. N. Wong. 1985. Generalized vector space model in information retrieval. In *Proceedings of the 8th ACM SIGIR Conference*, pages 18–25, Montreal.
Yarowsky, David. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994)*, pages 88–95, Las Cruces, NM.
Yuret, Deniz. 2004. Some experiments with a naive Bayes WSD system. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 265–268, Barcelona.
Zhao, Shubin and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 419–426, Ann Arbor, MI.