Kernel Combination for Word Sense Disambiguation

The IST Programme
Shared-Cost RTD
MEANING
Developing Multilingual Web-scale Language Technologies
Contract Number: IST-2001-34460

Workpackage: WP6
Deliverable: WP6.18 - 6N
Version: Final

Authors: Claudio Giuliano, Alfio Gliozzo, Carlo Strapparava - ITC-irst

Date: 19 January 2005

Contents

1 Introduction

In this report 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.

2 Semantic Domains and LSA

Domains are common areas of human discussion, such as economics, politics, law, science etc., which are at the basis of lexical coherence. A substantial part of the lexicon is composed by "domain words", that refer to concepts belonging to specific domains. In [Magnini *et al.*, 2002] it has been claimed that domain information provides generalized features at the paradigmatic level that are useful to discriminate among word senses.

The WordNet Domains[1] lexical resource is an extension of WordNet which provides such domain labels for all synsets [Magnini and Cavaglià, 2000]. About 200 domain labels were selected from a number of dictionaries and then structured in a taxonomy according to the Dewey Decimal Classification (DDC). The annotation methodology was mainly manual and took about 2 person years.

WordNet Domains has been proven a useful resource for WSD. However some aspects induced us to explore further developments. These issues are: (i) it is difficult to find an objective a-priori model for domains; (ii) the annotation procedure followed to develop WordNet Domains is very expensive, making hard the replicability of the lexical resource for other languages or domain specific sub-languages; (iii) the domain distinctions are rigid in WordNet Domains, while a more "fuzzy" association between domains and concepts is often more appropriate to describe term similarity.

In order to generalize the domain approach and to overcome these issues, we explored the direction of unsupervised learning on a large-scale corpus.

In particular, we followed the LSA approach [Deerwester *et al.*, 1990]. In LSA, term co-occurrences in the documents of the corpus are captured by means of a dimensionality reduction operated on the term-by-document matrix. The resulting LSA vectors can be exploited to estimate both term and document similarity. Regarding document similarity, Latent Semantic Indexing (LSI) is a technique that allows one to represent a document by a LSA vector. In particular, we used a variation of the *pseudo-document* methodology described in [Berry, 1992]. Each document can be represented in the LSA space by summing up the normalized LSA vectors of all the terms contained in it.

[1] WordNet Domains is freely available for research purposes at `wndomains.itc.it`

By exploiting LSA vectors for terms, it is possible to estimate domain vectors for the synsets of WordNet, in order to obtain similarity values between concepts that can be used for synset clustering and WSD. Thus, term and document vectors can be used instead of WordNet Domains for WSD and other applications in which term similarity and domain relevance estimation is required.

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

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

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 LSA for semantic domains (see Section 2) is exploited to define an explicit mapping $\mathcal{D} : \mathbb{R}^k \rightarrow \mathbb{R}^{k'}$ from the classical vector space model into the "domain" vector space model. The Domain Kernel is defined by

$$K_D(t_i, t_j) = \frac{\langle \mathcal{D}(t_i), \mathcal{D}(t_j) \rangle}{\sqrt{\langle \mathcal{D}(t_i), \mathcal{D}(t_j) \rangle \langle \mathcal{D}(t_i), \mathcal{D}(t_j) \rangle}} \quad (2)$$

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 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 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).

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^n_{Coll} is defined as a gap-weighted subsequence kernel applied to sequences of lemmata around the word l_0 to be disambiguated. In analogy we defined the PoS kernel K^n_{PoS}, by setting s to the sequence of PoSs around the word to be disambiguated.

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 1. 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.

4.1 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 1 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 1: 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 2. The results show that combining kernels significantly improves the performance of the system. Note that, differently from the other sections, the results presented here are evaluated by three-fold cross-validation on the training data.

	K_D	K_{BoW}	K_{PoS}	K_{Coll}	K_{wsd}	K'_{wsd}
F1	61.7	61.5	61.0	64.9	**66.8**	**69.9**

Table 2: The performance (F1) of each basic kernel and their combination for English lexical sample task (3-fold cross-validation on training data).

4.3 Portability and Performance

	MF	*Agreement*	*BEST*	K_{wsd}	K'_{wsd}	*DM+*	*BEST+*
English	55.2	67.3	72.9	69.7	**73.3**	3.6	0.4
Catalan	66.3	93.1	85.2	85.2	**89.0**	3.8	3.8
Italian	18.0	89.0	53.1	53.1	**61.3**	8.2	8.2
Spanish	67.7	85.3	84.2	84.2	**88.2**	4.0	4.0

Table 3: 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. $K'_{wsd} - K_{wsd}$), and *BEST+* (the improvement on the state-of-the-art, i.e. K'_{wsd} - BEST).

We evaluated the performance of K'_{wsd} and K_{wsd} on the lexical sample tasks described above. The results are showed in Table 3 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[2].

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

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

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 4, 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 4: Percentage of sense tagged examples required by K'_{wsd} to achieve the same performance of K_{wsd} with full training.

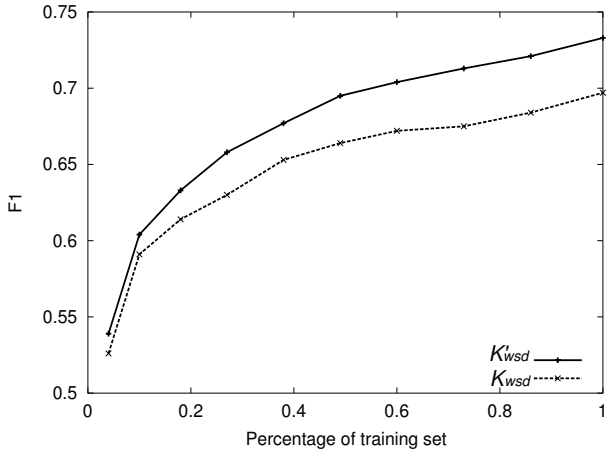

Figure 1: Learning curves for English lexical sample task.

5 Conclusion

In this report 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

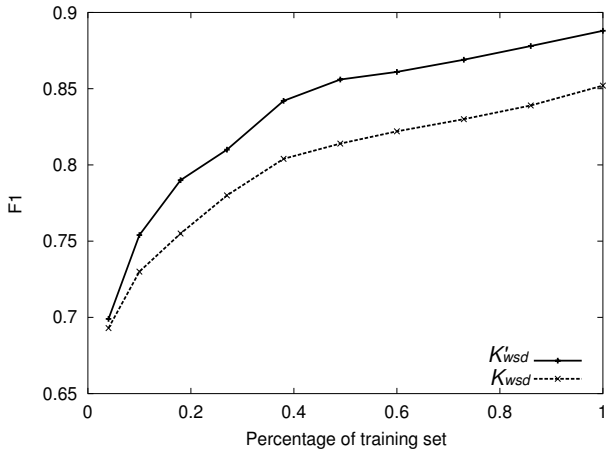

Figure 2: Learning curves for Catalan lexical sample task.

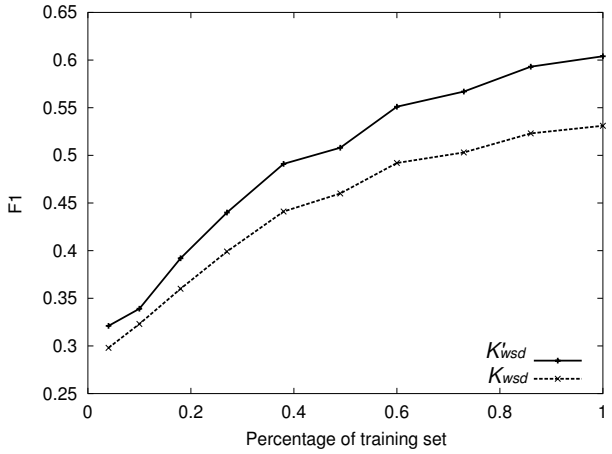

Figure 3: Learning curves for Italian lexical sample task.

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 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.

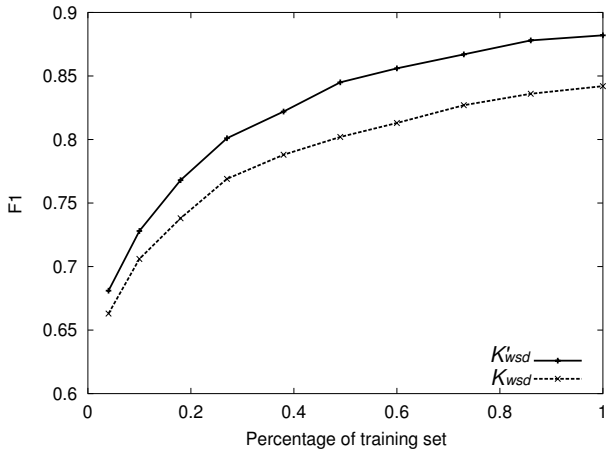

Figure 4: Learning curves for Spanish lexical sample task.

References

[Berry, 1992] M. Berry. Large-scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

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

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

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

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

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

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