Domain Kernels for Text Categorization

Alfio Gliozzo and **Carlo Strapparava**
ITC-Irst
via Sommarive, I-38050, Trento, ITALY
{gliozzo,strappa}@itc.it

Abstract

In this paper we propose and evaluate a technique to perform semi-supervised learning for Text Categorization. In particular 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 the Domain Kernel in two standard benchmarks for Text Categorization with good results, and we compared its performance with a kernel function that exploits a standard bag-of-words feature representation. The learning curves show that the Domain Kernel allows us to reduce drastically the amount of training data required for learning.

1 Introduction

Text Categorization (TC) deals with the problem of assigning a set of category labels to documents. Categories are usually defined according to a variety of topics (e.g. SPORT vs. POLITICS) and a set of hand tagged examples is provided for training. In the state-of-the-art TC settings supervised classifiers are used for learning and texts are represented by means of *bag-of-words*.

Even if, in principle, supervised approaches reach the best performance in many Natural Language Processing (NLP) tasks, in practice it is not always easy to apply them to concrete applicative settings. In fact, supervised systems for TC require to be trained a large amount of hand tagged texts. This situation is usually feasible only when there is someone (e.g. a big company) that can easily provide already classified documents to train the system.

In most of the cases this scenario is quite unpractical, if not infeasible. An example is the task of categorizing personal documents, in which the categories can be modified according to the user's interests: new categories are often introduced and, possibly, the available labeled training for them is very limited.

In the NLP literature the problem of providing large amounts of manually annotated data is known as the Knowledge Acquisition Bottleneck. Current research in supervised approaches to NLP often deals with defining methodologies and algorithms to reduce the amount of human effort required for collecting labeled examples.

A promising direction to solve this problem is to provide unlabeled data together with labeled texts to help supervision. In the Machine Learning literature this learning schema has been called *semi-supervised learning*. It has been applied to the TC problem using different techniques: co-training (Blum and Mitchell, 1998), EM-algorithm (Nigam et al., 2000), Transduptive SVM (Joachims, 1999b) and Latent Semantic Indexing (Zelikovitz and Hirsh, 2001).

In this paper we propose a novel technique to perform semi-supervised learning for TC. The underlying idea behind our approach is that lexical co-

Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL),
pages 56–63, Ann Arbor, June 2005. ©2005 Association for Computational Linguistics

herence (i.e. co-occurence in texts of semantically related terms) (Magnini et al., 2002) is an inherent property of corpora, and it can be exploited to help a supervised classifier to build a better categorization hypothesis, even if the amount of labeled training data provided for learning is very low.

Our proposal consists of defining a Domain Kernel and exploiting it inside a Support Vector Machine (SVM) classification framework for TC (Joachims, 2002). The Domain Kernel relies on the notion of Domain Model, which is a shallow representation for lexical ambiguity and variability. Domain Models can be acquired in an unsupervised way from unlabeled data, and then exploited to define a Domain Kernel (i.e. a generalized similarity function among documents)[1].

We evaluated the Domain Kernel in two standard benchmarks for TC (i.e. Reuters and 20Newsgroups), and we compared its performance with a kernel function that exploits a more standard Bag-of-Words (BoW) feature representation. The use of the Domain Kernel got a significant improvement in the learning curves of both tasks. In particular, there is a notable increment of the recall, especially with few learning examples. In addition, F1 measure increases by 2.8 points in the Reuters task at full learning, achieving the state-of-the-art results.

The paper is structured as follows. Section 2 introduces the notion of Domain Model and describes an automatic acquisition technique based on Latent Semantic Analysis (LSA). In Section 3 we illustrate the SVM approach to TC, and we define a Domain Kernel that exploits Domain Models to estimate similarity among documents. In Section 4 the performance of the Domain Kernel are compared with a standard bag-of-words feature representation, showing the improvements in the learning curves. Section 5 describes the previous attempts to exploit semi-supervised learning for TC, while section 6 concludes the paper and proposes some directions for future research.

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 $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ be a corpus, let $V = \{w_1, w_2, \dots, w_k\}$ be its vocabulary, let $\mathbf{T}$ be the $k \times n$ term-by-document matrix representing $\mathcal{T}$, 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 $\mathbf{R}^k$, in which the text $t_j \in \mathcal{T}$ 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.

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 Domain Model is composed by soft clusters of terms. Each cluster represents a semantic domain (Gliozzo et al., 2004), i.e. a set of terms that often co-occur in texts having similar topics. A Domain Model 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

Domain Models can be used to describe lexical ambiguity and variability. Lexical ambiguity is rep-

[1] The idea of exploiting a Domain Kernel to help a supervised classification framework, has been profitably used also in other NLP tasks such as word sense disambiguation (see for example (Strapparava et al., 2004)).

resented 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 Domain Model 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 $\mathcal{D} : \mathbf{R}^k \rightarrow \mathbf{R}^{k'}$, 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[2]

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

where $\mathbf{I^{IDF}}$ is a 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. Domain Vectors for texts are estimated by exploiting formula 1, while the Domain Vector $\vec{w'_i}$, 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 Domain Vectors 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.

In this work we propose the use of Latent Semantic Analysis (LSA) (Deerwester et al., 1990) to induce Domain Models 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 $\mathcal{T}$ 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[3]. Under this setting we define the domain matrix $\mathbf{D_{LSA}}$[4] as

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

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 The Domain Kernel

Kernel Methods are the state-of-the-art supervised framework for learning, and they have been successfully adopted to approach the TC task (Joachims, 1999a).

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. 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 a kernel function can work with any kernel-based algorithm, such as for example SVM.

During the learning phase SVMs assign a weight $\lambda_i \geq 0$ to any example $x_i \in X$. All the labeled instances x_i such that $\lambda_i > 0$ are called *support vectors*. The support vectors lie close to the best separating hyper-plane between positive and negative examples. New examples are then assigned to the class of its closest support vectors, according to equation 3.

[2] In (Wong et al., 1985) a similar schema is adopted to define a Generalized Vector Space Model, of which the Domain VSM is a particular instance.

[3] It is not clear how to choose the right dimensionality. In our experiments we used 400 dimensions.

[4] When $\mathbf{D_{LSA}}$ is substituted in Equation 1 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 adopted in Information Retrieval.

$$f(x) = \sum_{i=1}^{n} \lambda_i K(x_i, x) + \lambda_0 \quad (3)$$

The kernel function K returns the similarity between two instances in the input space X, and can be designed in order to capture the relevant aspects to estimate similarity, just by taking care of satisfying set of formal requirements, as described in (Schölkopf and Smola, 2001).

In this paper we define the Domain Kernel and we apply it to TC tasks. The Domain Kernel, denoted by K_D, can be exploited to estimate the topic similarity among two texts while taking into account the external knowledge provided by a Domain Model (see section 2). It is a variation of the Latent Semantic Kernel (Shawe-Taylor and Cristianini, 2004), in which a Domain Model is exploited to define an explicit mapping $\mathcal{D} : \mathbf{R}^k \rightarrow \mathbf{R}^{k'}$ 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. To be fully defined, the Domain Kernel requires a Domain Matrix $\mathbf{D}$. In principle, $\mathbf{D}$ can be acquired from any corpora by exploiting any (soft) term clustering algorithm. Anyway, we belive that adequate Domain Models for particular tasks can be better acquired from collections of documents from the same source. For this reason, for the experiments reported in this paper, we acquired the matrix $\mathbf{D_{LSA}}$, defined by equation 2, using the whole (unlabeled) training corpora available for each task, so tuning the Domain Model on the particular task in which it will be applied.

A more traditional approach to measure topic similarity among text consists of extracting BoW features and to compare them in a vector space. 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 Domain Model, so we can consider this setting as "purely" supervised, in which no external knowledge source is provided.

4 Evaluation

We compared the performance of both K_D and K_{BoW} on two standard TC benchmarks. In subsection 4.1 we describe the evaluation tasks and the preprocessing steps, in 4.2 we describe some algorithmic details of the TC system adopted. Finally in subsection 4.3 we compare the learning curves of K_D and K_{BoW}.

4.1 Text Categorization tasks

For the experiments reported in this paper, we selected two evaluation benchmarks typically used in the TC literature (Sebastiani, 2002): the *20newsgroups* and the *Reuters* corpora. In both the data sets we tagged the texts for part of speech and we considered only the noun, verb, adjective, and adverb parts of speech, representing them by vectors containing the frequencies of each disambiguated lemma. The only feature selection step we performed was to remove all the closed-class words from the document index.

20newsgroups. The *20Newsgroups* data set[5] is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. Some of the newsgroups are very closely related to each other (e.g. `comp.sys.ibm.pc.hardware` / `comp.sys.mac.hardware`), while others are highly unrelated (e.g. `misc.forsale` / `soc.religion.christian`). We removed cross-posts (duplicates), newsgroup-identifying headers (i.e. Xref, Newsgroups, Path, Followup-To, Date), and empty documents from the original corpus, so to obtain 18,941 documents. Then we randomly divided it into training (80%) and test (20%) sets, containing respectively 15,153 and 3,788 documents.

Reuters. We used the *Reuters-21578* collection[6], and we splitted it into training and test

[5] Available at `http://www.ai.mit.edu-/people/jrennie/20Newsgroups/`.

[6] Available at `http://kdd.ics.uci.edu/databases/-reuters21578/reuters21578.html`.

partitions according to the standard *ModAptè* split. It includes 12,902 documents for 90 categories, with a fixed splitting between training and test data. We conducted our experiments by considering only the 10 most frequent categories, i.e. `Earn, Acquisition, Money-fx, Grain, Crude, Trade, Interest, Ship, Wheat` and `Corn`, and we included in our dataset all the non empty documents labeled with at least one of those categories. Thus the final dataset includes 9295 document, of which 6680 are included in the training partition, and 2615 are in the test set.

4.2 Implementation details

As a supervised learning device, we used the SVM implementation described in (Joachims, 1999a). The Domain Kernel is implemented by defining an explicit feature mapping according to formula 1, and by normalizing each vector to obtain vectors of unitary length. All the experiments have been performed on the standard parameter settings, using a linear kernel.

We acquired a different Domain Model for each corpus by performing the SVD processes on the term-by-document matrices representing the whole training partitions, and we considered only the first 400 domains (i.e. $k' = 400$)[7].

As far as the *Reuters* task is concerned, the TC problem has been approached as a set of binary filtering problems, allowing the TC system to provide more than one category label to each document. For the *20newsgroups* task, we implemented a one-versus-all classification schema, in order to assign a single category to each news.

4.3 Domain Kernel versus BoW Kernel

Figure 1 and Figure 2 report the learning curves for both K_D and K_{BoW}, evaluated respectively on the *Reuters* and the *20newgroups* task. Results clearly show that K_D always outperforms K_{BoW}, especially when very limited amount of labeled data is provided for learning.

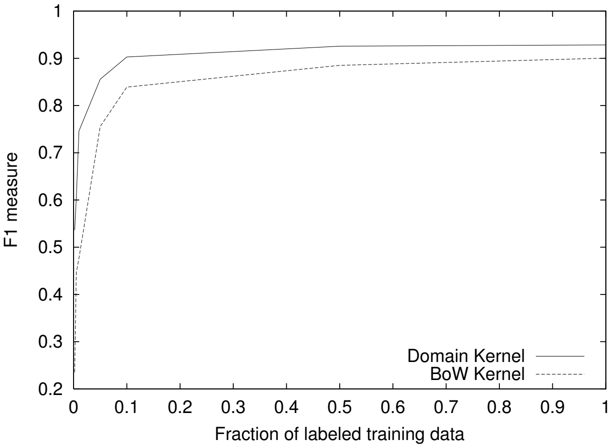

Figure 1: Micro-F1 learning curves for *Reuters*

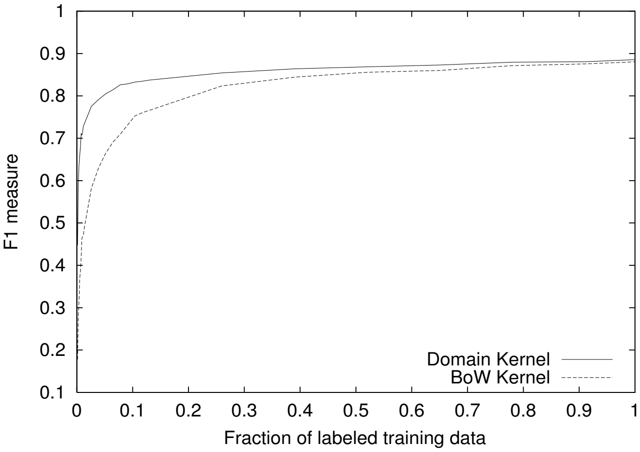

Figure 2: Micro-F1 learning curves for *20newsgroups*

Table 2 compares the performances of the two kernels at full learning. K_D achieves a better micro-F1 than K_{BoW} in both tasks. The improvement is particularly significant in the *Reuters* task (+ 2.8 %).

Tables 3 shows the number of labeled examples required by K_D and K_{BoW} to achieve the same micro-F1 in the *Reuters* task. K_D requires only 146 examples to obtain a micro-F1 of 0.84, while K_{BoW} requires 1380 examples to achieve the same performance. In the same task, K_D surpass the performance of K_{BoW} at full learning using only the 10% of the labeled data. The last column of the table shows clearly that K_D requires 90% less labeled data than K_{BoW} to achieve the same performances.

A similar behavior is reported in Table 4 for the

[7] To perform the SVD operation we adopted LIBSVDC, an optimized package for sparse matrix that allows to perform this step in few minutes even for large corpora. It can be downloaded from `http://tedlab.mit.edu/∼dr/SVDLIBC/`.

F1	*Domain Kernel*	*Bow Kernel*
Reuters	**0.928**	0.900
20newsgroups	**0.886**	0.880

Table 2: Micro-F1 with full learning

F1	*Domain Kernel*	*Bow Kernel*	*Ratio*
.54	**14**	267	5%
.84	**146**	1380	10%
.90	**668**	6680	10%

Table 3: Number of training examples needed by K_D and K_{BoW} to reach the same micro-F1 on the *Reuters* task

20newsgroups task. It is important to notice that the number of labeled documents is higher in this corpus than in the previous one. The benefits of using Domain Models are then less evident at full learning, even if they are significant when very few labeled data are provided.

Figures 3 and 4 report a more detailed analysis by comparing the micro-precision and micro-recall learning curves of both kernels in the *Reuters* task[8]. It is clear from the graphs that the main contribute of K_D is about increasing recall, while precision is similar in both cases[9]. This last result confirms our hypothesis that the information provided by the Domain Models allows the system to generalize in a more effective way over the training examples, allowing to estimate the similarity among texts even if they have just few words in common.

Finally, K_D achieves the state-of-the-art in the *Reuters* task, as reported in section 5.

5 Related Works

To our knowledge, the first attempt to apply the semi-supervised learning schema to TC has been reported in (Blum and Mitchell, 1998). Their co-training algorithm was able to reduce significantly the error rate, if compared to a strictly supervised classifier.

F1	*Domain Kernel*	*Bow Kernel*	*Ratio*
.50	**30**	500	6%
.70	**98**	1182	8%
.85	**2272**	7879	29%

Table 4: Number of training examples needed by K_D and K_{BoW} to reach the same micro-F1 on the *20newsgroups* task

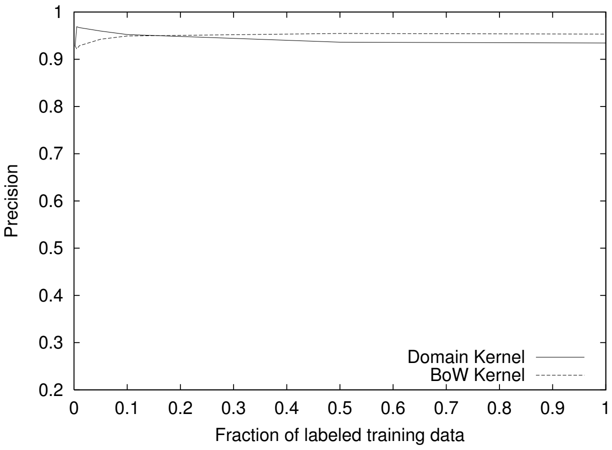

Figure 3: Learning curves for *Reuters* (Precision)

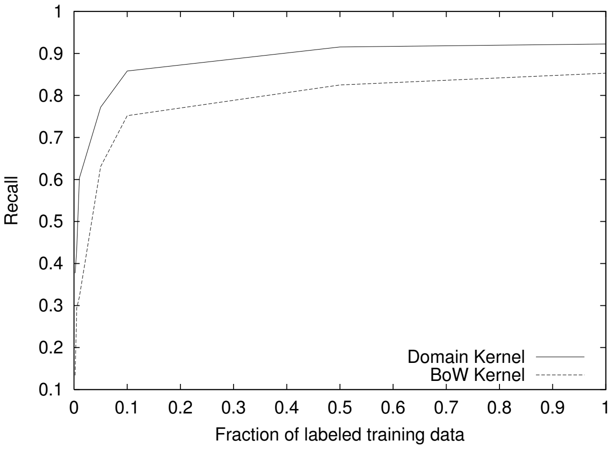

Figure 4: Learning curves for *Reuters* (Recall)

(Nigam et al., 2000) adopted an Expectation Maximization (EM) schema to deal with the same problem, evaluating extensively their approach on several datasets. They compared their algorithm with a standard probabilistic approach to TC, reporting substantial improvements in the learning curve.

[8] For the 20-newsgroups task both micro-precision and micro-recall are equal to micro-F1 because a single category label has been assigned to every instance.

[9] It is worth noting that K_D gets a F1 measure of 0.54 (Precision/Recall of 0.93/0.38) using just 14 training examples, suggesting that it can be profitably exploited for a bootstrapping process.

A similar evaluation is also reported in (Joachims, 1999b), where a transduptive SVM is compared to a state-of-the-art TC classifier based on SVM. The semi-supervised approach obtained better results than the standard with few learning data, while at full learning results seem to converge.

(Bekkerman et al., 2002) adopted a SVM classifier in which texts have been represented by their associations to a set of Distributional Word Clusters. Even if this approach is very similar to ours, it is not a semi-supervised learning schema, because authors did not exploit any additional unlabeled data to induce word clusters.

In (Zelikovitz and Hirsh, 2001) background knowledge (i.e. the unlabeled data) is exploited together with labeled data to estimate document similarity in a Latent Semantic Space (Deerwester et al., 1990). Their approach differs from the one proposed in this paper because a different categorization algorithm has been adopted. Authors compared their algorithm with an EM schema (Nigam et al., 2000) on the same dataset, reporting better results only with very few labeled data, while EM performs better with more training.

All the semi-supervised approaches in the literature reports better results than strictly supervised ones with few learning, while with more data the learning curves tend to converge.

A comparative evaluation among semi-supervised TC algorithms is quite difficult, because the used data sets, the preprocessing steps and the splitting partitions adopted affect sensibly the final results. Anyway, we reported the best F1 measure on the Reuters corpus: to our knowledge, the state-of-the-art on the 10 top most frequent categories of the ModApte split at full learning is F1 92.0 (Bekkerman et al., 2002) while we obtained 92.8. It is important to notice here that this results has been obtained thanks to the improvements of the Domain Kernel. In addition, on the *20newsgroups* task, our methods requires about 100 documents (i.e. five documents per category) to achieve 70% F1, while both EM (Nigam et al., 2000) and LSI (Zelikovitz and Hirsh, 2001) requires more than 400 to achieve the same performance.

6 Conclusion and Future Works

In this paper a novel technique to perform semi-supervised learning for TC has been proposed and evaluated. We defined a Domain Kernel that allows us to improve the similarity estimation among documents by exploiting Domain Models. Domain Models are acquired from large collections of non annotated texts in a totally unsupervised way.

An extensive evaluation on two standard benchmarks shows that the Domain Kernel allows us to reduce drastically the amount of training data required for learning. In particular the recall increases sensibly, while preserving a very good accuracy. We explained this phenomenon by showing that the similarity scores evaluated by the Domain Kernel takes into account both variability and ambiguity, being able to estimate similarity even among texts that do not have any word in common.

As future work, we plan to apply our semi-supervised learning method to some concrete applicative scenarios, such as user modeling and categorization of personal documents in mail clients. In addition, we are going deeper in the direction of semi-supervised learning, by acquiring more complex structures than clusters (e.g. synonymy, hyperonymy) to represent domain models. Furthermore, we are working to adapt the general framework provided by the Domain Models to a multilingual scenario, in order to apply the Domain Kernel to a Cross Language TC task.

Acknowledgments

This work has been partially supported by the ONTOTEXT (From Text to Knowledge for the Semantic Web) project, funded by the Autonomous Province of Trento under the FUP-2004 research program.

References

R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. 2002. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 1:1183–1208.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers.*

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*, 18:275–299.

T. Joachims. 1999a. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods: support vector learning*, chapter 11, pages 169 – 184. MIT Press, Cambridge, MA, USA.

T. Joachims. 1999b. Transductive inference for text classification using support vector machines. In *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209. Morgan Kaufmann Publishers, San Francisco, US.

T. Joachims. 2002. *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers.

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.

K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

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

B. Schölkopf and A. J. Smola. 2001. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

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

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*.

S. Zelikovitz and H. Hirsh. 2001. Using LSI for text classification in the presence of background text. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management*, pages 113–118, Atlanta, US. ACM Press, New York, US.