A Semantic Kernel to classify texts with very few training examples

Roberto Basili BASILI@INFO.UNIROMA2.IT
Marco Cammisa CAMMISA@INFO.UNIROMA2.IT
Alessandro Moschitti MOSCHITTI@INFO.UNIROMA2.IT
Department of Computer Science, Systems and Production,
University of Rome "Tor Vergata",
Via del Politecnico 1, 00133 Rome, Italy

Abstract

Web-mediated access to distributed information is a complex problem. Before any learning can start, Web objects (e.g. texts) have to be detected and filtered accurately. In this perspective, text categorization is a useful device to filter out irrelevant evidence before other learning processes take place on huge sources of candidate information. The drawback is the need of a large number of training documents. One way to reduce such number relates to the use of more effective document similarities based on prior knowledge. Unfortunately, previous work has shown that such information (e.g. WordNet) causes the decrease of retrieval accuracy.

In this paper we propose kernel functions to add prior knowledge to learning algorithms for document classification. Such kernels use a term similarity measure based on the WordNet hierarchy. The kernel trick is used to implement such space in a balanced and statistically coherent way. Cross-validation results show the benefit of the approach for the Support Vector Machines when few training examples are available.

1. Introduction

Web-mediated access to distributed information is a complex problem. Before any learning can start, Web objects (e.g. texts) have to be detected and filtered accurately. In this perspective, text categorization (TC) is a useful device to filter out irrelevant evidence before other learning processes take place on huge sources of candidate information. To apply TC in Web search, methods based on small number of examples should be preferred. As such number decreases the classification accuracy decreases as well, thus, to mitigate this problem, most of the research efforts have been directed in enriching the document representation by using term clustering (*term generalization*) or adding compound terms (*term specification*). These studies are based on the assumption that the similarity between two documents can be expressed as the similarity between pairs of matching terms. Following this idea, term clustering methods based on corpus term distributions or on external (to the target corpus) prior knowledge (e.g. provided by WordNet) were used to improve the basic term matching.

An example of statistical clustering is given in (Bekkerman et al., 2001). A feature selection technique, which clusters similar features/words, called the Information Bottleneck (IB), was applied to Text Categorization (TC). Such cluster based representation outperformed the simple *bag-of-words* on only one out of the three experimented collections. The effective use of external prior knowledge is even more difficult since no attempt has ever been successful to improve document retrieval or text classification accuracy, (e.g. see (Smeaton, 1999; Sussna, 1993; Voorhees, 1993; Voorhees, 1994; Moschitti & Basili, 2004)).

The main problem of term cluster based representations seems the unclear nature of the relationship between the word and the cluster information levels. Although (semantic) clusters tend to improve the system Recall, simple terms are, on a large scale, more accurate (e.g. (Moschitti & Basili, 2004)). To overcome this problem the hybrid spaces containing terms and clusters were experimented (e.g. (Scott & Matwin, 1999)) but the results, again, showed that the mixed statistical distributions of clusters and terms impact

Appearing in *W4: Learning in Web Search, at the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

either marginally or even negatively on the overall accuracy.

In (Voorhees, 1993; Smeaton, 1999), clusters of synonymous terms as defined in WordNet (WN) (Fellbaum, 1998) were used for document retrieval. The results showed that the misleading information due to the wrong choice of the local term senses causes the overall accuracy to decrease. Word sense disambiguation (WSD) was thus applied beforehand by indexing the documents by means of disambiguated senses, i.e. synset codes (Smeaton, 1999; Sussna, 1993; Voorhees, 1993; Voorhees, 1994; Moschitti & Basili, 2004). However, even the state-of-the-art methods for WSD did not improve the accuracy because of the inherent noise introduced by the disambiguation mistakes. The above studies suggest that term clusters decrease the precision of the system as they force weakly related terms or unrelated terms (in case of disambiguation errors) to give a contribution in the similarity function. The successful introduction of prior external knowledge relies on the solution of the above problem.

In this paper, a model to introduce the semantic lexical knowledge contained in the WN hierarchy in a supervised text classification task has been proposed. Intuitively, the main idea is that the documents d are represented through the set of all pairs $< t, t' >$ originating by the terms $t \in d$ and all the words $t' \in V$, e.g. the WN's nouns. When the similarity between two documents is evaluated, their matching pairs are used to account for the final score. The weight given to each term pair is proportional to the similarity that the two terms have in WN. Thus, the term t of the first document contributes to the document similarity according to its relatedness with any of the terms of the second document and the prior external knowledge, provided by WN, quantifies the single term to term relatedness. Such approach has two advantages: (a) we obtain a well defined space which supports the similarity between terms of different surface forms based on external knowledge and (b) we avoid to explicitly define term or sense clusters which inevitably introduce noise.

The class of spaces which embeds the above pair information may be composed by $O(|V|^2)$ dimensions. If we consider only the WN nouns (about 10^5), our space contains about 10^{10} dimensions which is not manageable by most part of the learning algorithms. Kernel methods, can solve this problem as they allow us to use an implicit space representation in the learning algorithms. Among other Support Vector Machines (SVMs) (Vapnik, 1995) are kernel based learners which achieve high accuracy in presence of many irrelevant features. This is another important property for our approach as we leave the selection of the informative pairs to the SVM learning.

Moreover, as we believe that the prior knowledge in TC is not so useful when there is a sufficient amount of training documents, we experimented our model in poor training conditions (e.g. less equal than 20 documents for each category). The improvement in the accuracy, observed on the classification of the well known Reuters and 20 NewsGroups corpora, shows that our document similarity model is very promising for general IR tasks: unlike previous attempts, it makes sense of the adoption of semantic external resources (i.e. WN) in IR.

Section 2 introduces the WordNet-based term similarity. Section 3 defines the new document similarity measure, the kernel function and its use within SVMs. Section 4 presents the comparative results between the traditional linear and the WN-based kernels within SVMs. In Section 5 comparative discussion against the related IR literature is carried out. Finally Section 6 derives the conclusions.

2. Term similarity based on general knowledge

In IR, any similarity metric in the vector space models is driven by lexical matching. When small training material is available, few words can be effectively used and the resulting document similarity metrics are very weak. Semantic generalizations overcome data sparseness problems in IR as contributions from different but semantically similar words are made available.

Methods for the induction of semantically inspired word clusters have been widely used in language modeling and lexical acquisition tasks (e.g. (Clark & Weir, 2002)). The main resource employed in most works is WordNet (Fellbaum, 1998) which contains three subhierarchies: for nouns, verbs and adjectives. Each hierarchy represents lexicalized concepts (or senses) organized according to an "*is-a-kind-of*" relation. A concept s is described by a set of words $syn(s)$ called *synset*. The words $w \in syn(s)$ are synonyms according to the sense s.

For example, the words *line*, *argumentation*, *logical argument* and *line of reasoning* describe a synset which expresses the methodical process of logical reasoning (e.g. "*I can't follow your line of reasoning*"). Each word/term may be lexically related to more than one synset depending on the senses that it assumes. The word *line* is also present in the synset *line*, *dividing*

line, *demarcation* and *contrast*, to emphasize that a *line* denotes a conceptual separation or demarcation (e.g. "*there is a narrow line between sanity and insanity*").

In the next section we define a term similarity measure based on the WN noun hierarchy. Such hierarchy is a direct acyclic graph[1] in which the edges establish the *direct_isa* relations between two synsets.

2.1. The Conceptual Density

The automatic use of WordNet for NLP and IR tasks has proved to be very complex. First, how the topological distance among senses is related to their corresponding conceptual distance is unclear. The pervasive lexical ambiguity is also problematic as it impacts on the measure of conceptual distances between word pairs. Second, the approximation of a set of concepts by means of their generalization in the hierarchy implies a conceptual loss that affects the target IR (or NLP) tasks. For example, *black* and *white* are *colors* but are also *chess pieces* and this impacts on the similarity score that should be used in IR applications. Attempts to solve the above problems relates to *cuts* in the hierarchy (e.g. (Li & Abe, 1998; Resnik, 1997)) by using corpus statistics. For several tasks (e.g. in TC) this is unsatisfactory: different contexts of the same corpus (e.g. documents) may require different generalizations of the same word as they independently impact on the document similarity.

On the contrary, the *Conceptual Density* (CD) (Agirre & Rigau, 1996) is a flexible semantic similarity which depends on the generalizations of word senses not referring to any fixed level of the hierarchy. Its formal definition is given in what follows.

We denote by $\bar{s}$ the set of nodes of the hierarchy rooted in the synset s, i.e. $\{c \in S | c \; isa \; s\}$, where S is the set of WN synsets. By definition $\forall s \in S, s \in \bar{s}$. CD makes a guess about the proximity of the senses, s_1 and s_2, of two words u_1 and u_2, according to the information expressed by the minimal subhierarchy, $\bar{s}$, that includes them. Let S_i be the set of generalizations for at least one sense s_i of the word u_i, i.e. $S_i = \{s \in S | s_i \in \bar{s}, u_i \in syn(s_i)\}$. The CD of u_1 and u_2 is:

$$CD(u_1, u_2) = \begin{cases} 0 & \textbf{iff } S_1 \cap S_2 = \emptyset \\ max_{s \in S_1 \cap S_2} \frac{\sum_{i=0}^{h} (\mu(\bar{s}))^i}{|\bar{s}|} & \textbf{otherwise} \end{cases} \quad (1)$$

[1] As only the 1% of its nodes own more than one parent in the graph, most of the techniques assume the hierarchy to be a tree, and treat the few exception heuristically.

where:

- $S_1 \cap S_2$ is the set of WN shared generalizations (i.e. the common hypernyms) for u_1 and u_2
- $\mu(\bar{s})$ is the average number of children per node (i.e. the branching factor) in the sub-hierarchy $\bar{s}$. $\mu(\bar{s})$ depends on WordNet and in some cases its value can approach 1.
- h is the depth of the *ideal tree* whose leaves are only the two senses s_1 and s_2 and the average branching factor is $\mu(\bar{s})$. This value is actually estimated by:

$$h = \begin{cases} \lfloor log_{\mu(\bar{s})} 2 \rfloor & \textbf{iff } \mu(\bar{s}) \neq 1 \\ 2 & \textbf{otherwise} \end{cases} \quad (2)$$

  In cases $\mu(s)$ is exactly 1 the above equation assigns 2 to h.
- $|\bar{s}|$ is the number of nodes in the sub-hierarchy $\bar{s}$. This value is statically measured on WN and it is a negative bias for the higher level of generalizations (i.e. larger $\bar{s}$).

CD models the semantic distance as the density of the generalizations $s \in S_1 \cap S_2$. Such *density* is the ratio between the number of nodes of the *ideal tree* and $|\bar{s}|$. The ideal tree should (a) link the two senses/nodes s_1 and s_2 with the minimal number of edges (isa-relations) and (b) maintain the same branching factor (bf) observed in $\bar{s}$. In other words, this tree provides the minimal number of nodes (and isa-relations) sufficient to connect s_1 and s_2 according to the topological structure of $\bar{s}$. For example, if $\bar{s}$ has a bf of 2 the ideal tree connects the two senses with a single node (their father). If the bf is 1.5, to replicate it, the ideal tree must contain 4 nodes, i.e. the grandfather which has a bf of 1 and the father which has bf of 2 for an average of 1.5. When bf is 1 the Eq. 1 degenerates to the inverse of the number of nodes in the path between s_1 and s_2, i.e. the simple proximity measure used in (Siolas & d'Alch Buc, 2000).

It is worth noting that for each pair $CD(u_1, u_2)$ determines the similarity according to *the closest lexical senses*, s_1, $s_2 \in \bar{s}$: the remaining senses of u_1 and u_2 are irrelevant, with a resulting semantic disambiguation side effect. The CD properties seem appealing to define similarity measures between any term pairs in IR models. As the high number of such pairs increases the computational complexity of the target learning algorithm, efficient approaches are needed. The next section describes how kernel methods can make practical the use of the Conceptual Density in Text Categorization.

3. A WordNet Kernel for document similarity

Term similarities are used to design document similarities which are the core functions of most TC algorithms. The term similarity proposed in Eq. 1 is valid for all term pairs of a target vocabulary and has two main advantages: (1) the relatedness of each term occurring in the first document can be computed against *all* terms in the second document, i.e. all different pairs of similar (not just identical) tokens can contribute and (2) if we use all term pair contributions in the document similarity we obtain a measure consistent with the term probability distributions, i.e. the sum of all term contributions does not penalize or emphasize arbitrarily any subset of terms. The next section presents more formally the above idea.

3.1. A *semantic* vector space

Given two documents d_1 and $d_2 \in D$ (the document-set) we define their similarity as:

$$K(d_1, d_2) = \sum_{w_1 \in d_1, w_2 \in d_2} (\lambda_1 \lambda_2) \times \sigma(w_1, w_2) \qquad (3)$$

where λ_1 and λ_2 are the weights of the words (features) w_1 and w_2 in the documents d_1 and d_2, respectively and σ is a term similarity function, e.g. the conceptual density defined in Section 2. To prove that Eq. 3 is a valid kernel is enough to show that it is a specialization of the general definition of convolution kernels formalized in (Haussler, 1999). Hereafter, we report such definition: let $X, X_1, .., X_m$ be separable metric spaces, $x \in X$ a structure and $\vec{x} = x_1, ..., x_m$ its parts, where $x_i \in X_i \forall i = 1, .., m$. Let R be a relation on the set $X \times X_1 \times .. \times X_m$ such that $R(\vec{x}, x)$ holds if $\vec{x}$ are the parts of x. We indicate with $R^{-1}(x)$ the set $\{\vec{x} : R(\vec{x}, x)\}$. Given two objects x and $y \in X$ their similarity $K(x, y)$ is defined as:

$$K(x, y) = \sum_{\vec{x} \in R^{-1}(x)} \sum_{\vec{y} \in R^{-1}(y)} \prod_{i=1}^{m} K_i(x_i, y_i) \qquad (4)$$

If we consider X as the document set (i.e. $D = X$), $m = 1$ and $X_1 = V$ (i.e. the vocabulary of our target document corpus) we derive that: $x = d$ (i.e. a document), $\vec{x} = x_1 = w \in V$ (i.e. a word which is a part of the document d) and $R^{-1}(d)$ is the set of words in the document d. As $\prod_{i=1}^{m} K_i(x_i, y_i) = K_1(x_1, y_1)$, we can define $K_1(x_1, y_1) = K(w_1, w_2) = (\lambda_1 \lambda_2) \times \sigma(w_1, w_2)$ to obtain exactly the Eq. 3.

The above equation can be used in support vector machines as illustrated by the next section.

3.2. Support Vector Machines and Kernel methods

Given the vector space in $\mathbb{R}^\eta$ and a set of positive and negative points, SVMs classify vectors according to a separating hyperplane, $H(\vec{x}) = \vec{\omega} \cdot \vec{x} + b = 0$, where $\vec{x}$ and $\vec{\omega} \in \mathbb{R}^\eta$ and $b \in \mathbb{R}$ are learned by applying the *Structural Risk Minimization principle* (Vapnik, 1995). From the kernel theory we have that:

$$H(\vec{x}) = \Big(\sum_{h=1..l} \alpha_h \vec{x_h} \Big) \cdot \vec{x} + b = \sum_{h=1..l} \alpha_h \vec{x}_h \cdot \vec{x} + b =$$

$$\sum_{h=1..l} \alpha_h \phi(d_h) \cdot \phi(d) + b = \sum_{h=1..l} \alpha_h K(d_h, d) + b \qquad (5)$$

where, d is a classifying document and d_h are all the l training instances, projected in $\vec{x}$ and $\vec{x}_h$ respectively. The product $K(d, d_h) = <\phi(d) \cdot \phi(d_h)>$ is the *Semantic WN-based Kernel* (SK) function associated with the mapping ϕ.

Eq. 5 shows that to evaluate the separating hyperplane in $\mathbb{R}^\eta$ we do not need to evaluate the entire vector $\vec{x_h}$ or $\vec{x}$. Actually, we do not know even the mapping ϕ and the number of dimensions, η. As it is sufficient to compute $K(d, d_h)$, we can carry out the learning with Eq. 3 in the $\mathbb{R}^n$, avoiding to use the explicit representation in the $\mathbb{R}^\eta$ space. The real advantage is that we can consider only the word pairs associated with non-zero weights, i.e. we can use a sparse vector computation. Additionally, to have a uniform score across different document size, the kernel function can be normalized as follows: $\frac{SK(d_1, d_2)}{\sqrt{SK(d_1, d_1) \cdot SK(d_2, d_2)}}$

4. Experiments

The use of WordNet (WN) in the term similarity function introduces a prior knowledge whose impact on the Semantic Kernel (SK) should be experimentally assessed. The main goal is to compare the traditional Vector Space Model kernel against SK, both within the Support Vector learning algorithm.

The high complexity of the SK limits the size of the experiments that we can carry out in a feasible time. Moreover, we are not interested to large collections of training documents as in these training conditions the simple *bag-of-words* models are in general very effective, i.e. they seem to model well the document similarity needed by the learning algorithms. Thus, we carried out the experiments on small subsets of the

20NewsGroups[2] (20NG) and the *Reuters-21578*[3] corpora to simulate critical learning conditions.

4.1. Experimental set-up

For the experiments, we used the SVM-light software (Joachims, 1999) (available at `svmlight.joachims.org`) with the default linear kernel on the token space (adopted as the baseline evaluations). For the SK evaluation we implemented the Eq. 3 with $\sigma(\cdot,\cdot) = CD(\cdot,\cdot)$ (Eq. 1) inside SVM-light. As CD is sensitive only to nouns we detected them by means of a part of speech (POS) tagger. Nevertheless, given the importance of verbs, adjectives and numerical features for TC, we included them in the pair space by assigning a null value to the pairs made by different tokens. As the POS-tagger could introduce errors, we alternatively detected nouns by simply looking-up in WN, i.e. any word is considered as a noun if it is included in the noun WN hierarchy. This may be considered a rough approximation but it has the benefit to recover other useful information by including the similarity between the verb nominalizations and the other nouns, e.g. *to drive* like *drive* has a synset in common with *parkway*.

For the evaluations, we applied a careful SVM parameterization: a preliminary investigation suggested that the trade-off (between the training-set error and margin, i.e. c option in SVM-light) parameter optimizes the F_1 measure for values in the range [0.02,0.32][4]. We noted also that the cost-factor parameter (i.e. j option) is not critical, i.e. a value of 10 always optimizes the accuracy. The feature selection techniques and the weighting schemes were not applied in our experiments as they cannot be accurately estimated from the small available training data.

The classification performance was evaluated by means of the F_1 measure[5] for the single category and the MicroAverage for the final classifier pool (Yang, 1999). Given the high computational complexity of SK we selected 8 categories from the 20NG[6] and 8 from the Reuters corpus[7]

[2]Available at `www.ai.mit.edu/people/jrennie/20Newsgroups/`.

[3]The Apté split available at `kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

[4]We used all the values from 0.02 to 0.32 with step 0.02.

[5]F_1 assigns equal importance to Precision P and Recall R, i.e. $F_1 = \frac{2P \cdot R}{P+R}$.

[6]We selected the 8 most different categories (in terms of their content) i.e. *Atheism*, *Computer Graphics*, *Misc Forsale*, *Autos*, *Sport Baseball*, *Medicine*, *Talk Religions* and *Talk Politics*.

[7]We selected the 8 largest categories, i.e. *Acquisition*, *Crude*, *Earn*, *Grain*, *Interest*, *Money-fx*, *Trade* and *Wheat*.

To derive statistically significant results with few training documents, for each corpus, we randomly selected 10 different samples from the 8 categories. We trained the classifiers on one sample, parameterized on a second sample and derived the measures on the other 8. By rotating the training sample, we obtained 80 different measures for each model. The size of the samples ranges from 24 to 160 documents depending on the target experiment.

4.2. Cross validation results

The SK (Eq. 3) was compared with the linear kernel which obtained the best F_1 measure in (Joachims, 1999). Table 1 reports the first comparative results for 8 categories of 20NG on 40 training documents. The results are expressed as the *Mean* and the *Std. Dev.* over 80 runs. The F_1 are reported in Column 2 for the linear kernel, i.e. *bow*, in Column 3 for SK without applying POS information and in Column 4 for SK with the use of POS information (SK-POS). The last row shows the MicroAverage performance for the above three models on all 8 categories. We note that SK improves *bow* of 3%, i.e. 34.3% vs. 31.5% and that the POS information reduces the improvement of SK, i.e. 33.5% vs. 34.3%.

Category	*bow*	*SK*	*SK*-POS
Atheism	29.5±19.8	32.0±16.3	25.2±17.2
Comp.Graph	39.2±20.7	39.3±20.8	29.3±21.8
Misc.Forsale	61.3±17.7	51.3±18.7	49.5±20.4
Autos	26.2±22.7	26.0±20.6	33.5±26.8
Sport.Baseb.	32.7±20.1	36.9±22.5	41.8±19.2
Sci.Med	26.1±17.2	18.5±17.4	16.6±17.2
Talk.Relig.	23.5±11.6	28.4±19.0	27.6±17.0
Talk.Polit.	28.3±17.5	30.7±15.5	30.3±14.3
MicroAvg. F_1	31.5±4.8	34.3±5.8	33.5±6.4

Table 1. Performance of the linear and Semantic Kernel with 40 training documents over 8 categories of 20NewsGroups collection.

Category	24 docs		160 docs	
	bow	*SK*	*bow*	*SK*
Acq.	55.3±18.1	50.8±18.1	86.7±4.6	84.2±4.3
Crude	3.4±5.6	3.5±5.7	64.0±20.6	62.0±16.7
Earn	64.0±10.0	64.7±10.3	91.3±5.5	90.4±5.1
Grain	45.0±33.4	44.4±29.6	69.9±16.3	73.7±14.8
Interest	23.9±29.9	24.9±28.6	67.2±12.9	59.8±12.6
Money-fx	36.1±34.3	39.2±29.5	69.1±11.9	67.4±13.3
Trade	9.8±21.2	10.3±17.9	57.1±23.8	60.1±15.4
Wheat	8.6±19.7	13.3±26.3	23.9±24.8	31.2±23.0
Mic.Avg.	37.2±5.9	41.7±6.0	75.9±11.0	77.9±5.7

Table 2. Performance of the linear and Semantic Kernel with 24 and 160 training documents over 8 categories of the Reuters corpus.

To verify the hypothesis that WN information is useful

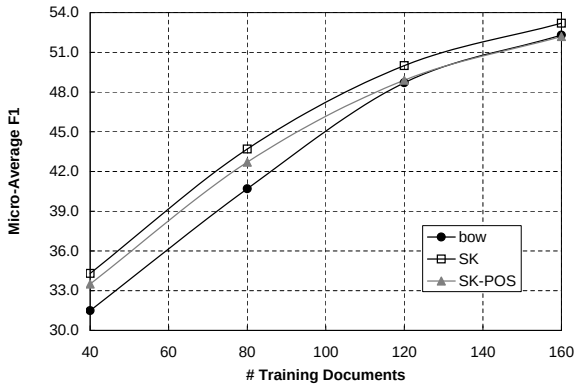

Figure 1. MicroAverage F_1 of SVMs using *bow*, SK and SK-POS kernels over the 8 categories of 20NewsGroups.

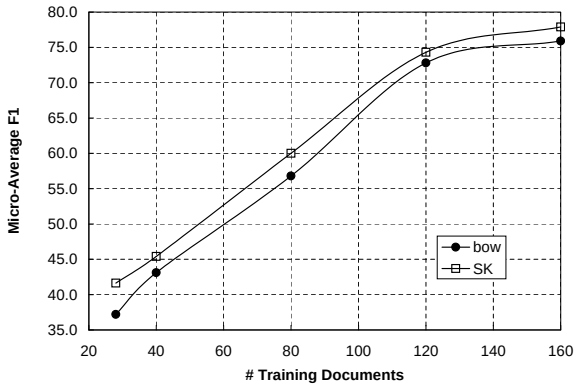

Figure 2. MicroAverage F_1 of SVMs using *bow* and SK over the 8 categories of the Reuters corpus.

in low training data conditions we repeated the evaluation over the 8 categories of Reuters with samples of 24 and 160 documents, respectively. The results reported in Table 2 shows that (1) again SK improves *bow* (41.7% - 37.2% = 4.5%) and (2) as the number of documents increases the improvement decreases (77.9% - 75.9% = 2%). It is worth noting that the standard deviations tend to assume high values. However, such variability does not affect the confidence test on the SK superiority. To verify that SK improves *bow*, we evaluated the Std. Dev. of the difference, d, between the MicroAverage F_1 of SK and the MicroAverage F_1 of *bow* over the samples. In relation to the Table 2 experiment, we obtained that the mean and the Std. Dev. of d on the 80 test samples of 24 documents are 4.53 and 6.57, respectively. We tested the hypothesis that *bow* has a higher or equal MicroAverage F_1 than SK, i.e. $d \leq 0$. Accordigly, the maximum value of the population average μ cannot be higher than 0, thus we tried the hypothesis $\mu = 0$. By using a Normal Distribution, d is in the range [-∞,μ+2.13] at a confidence level of 99.5%. Since the mean of the MicroAverage trough the samples (4.53) is not in such interval, we should reject such hypothesis.

The above findings confirm that SK outperforms the *bag-of-words* kernel in critical learning conditions as the semantic contribution of the SK recovers useful information. To complete this study we carried out experiments with samples of different size, i.e. 3, 5, 10, 15 and 20 documents for each category. Figures 1 and 2 show the learning curves for 20NG and Reuters corpora. Each point refers to the average on 80 samples.

As expected the improvement provided by SK decreases when more training data is available. However, the improvement is not negligible yet. The SK model (without POS information) preserves about 2-3% of improvement with 160 training documents. The matching allowed between noun-verb pairs still captures semantic information which is useful for topic detection. In particular, during the similarity estimation, each word activates 60.05 pairs on average. This is particularly useful to increase the amount of information available to the SVMs.

Finally, we carried out some experiments with 160 Reuters documents by discarding the string matching from SK. Only words having different surface forms were allowed to give contributions to the Eq. 3.

The interesting outcome is that SK converges to a MicroAverage F_1 measure of 56.4% (compare with Table 2). This shows that the word similarity provided by WN is consistent and effective for TC.

5. Related Work

The IR studies in this area focus on the term similarity models to embed statistical and external knowledge in document similarity.

In (Kontostathis & Pottenger, 2002) a *Latent Semantic Indexing* analysis was used for term clustering. Such approach assumes that values x_{ij} in the transformed term-term matrix represents the similarity (> 0) and anti-similarity between terms i and j. By extension, a negative value represents an anti-similarity between i and j enabling both positive and negative clusters of terms. Evaluation of query expansion techniques showed that positive clusters can improve Recall of about 18% for the *CISI* collection, 2.9% for *MED* and 3.4% for *CRAN*. Furthermore, the negative clusters, when used to prune the result set, improve the precision.

The use of external semantic knowledge seems to be

more problematic in IR. In (Smeaton, 1999), the impact of semantic ambiguity on IR is studied. A WN-based semantic similarity function between noun pairs is used to improve indexing and document-query matching. However, the WSD algorithm had a performance ranging between 60-70%, and this made the overall semantic similarity not effective.

Other studies using semantic information for improving IR were carried out in (Sussna, 1993) and (Voorhees, 1993; Voorhees, 1994). Word semantic information was here used for text indexing and query expansion, respectively. In (Voorhees, 1994) it is shown that semantic information derived directly from WN without a priori WSD produces poor results.

The latter methods are even more problematic in TC (Moschitti & Basili, 2004). Word senses tend to systematically correlate with the positive examples of a category. Different categories are better characterized by different words rather than different senses. Patterns of lexical co-occurrences in the training data seem to suffice for automatic disambiguation. (Scott & Matwin, 1999) use WN senses to replace simple words without word sense disambiguation and small improvements are derived only for a small corpus. The scale and assessment provided in (Moschitti & Basili, 2004) (3 corpora using cross-validation techniques) showed that even the accurate disambiguation of WN senses (about 80% accuracy on nouns) did not improve TC.

In (Siolas & d'Alch Buc, 2000) was proposed an approach similar to the one presented in this article. A term proximity function is used to design a kernel able to semantically smooth the similarity between two document terms. Such semantic kernel was designed as a combination of the Radial Basis Function (RBF) kernel with the term proximity matrix. Entries in this matrix are inversely proportional to the length of the WN hierarchy path linking the two terms. The performance, measured over the 20NewsGroups corpus, showed an improvement of 2% over the *bag-of-words*. The main differences with our approach are: first, the term proximity is not fully sensitive to the information of the WN hierarchy. For example, if we consider pairs of equidistant terms, the nearer to the WN top level a pair is the lower similarity it should receive, e.g. *Sky* and *Location* (hyponyms of *Entity*) should not accumulate similarity like *knife* and *gun* (hyponyms of *weapon*). Measures, like CD, that deal with this problem have been widely proposed in literature (e.g. (Resnik, 1997)) and should be always applied. Second, as our main goal was the study of the CD information in document retrieval/categorization scenario, our kernel function was based on the simple CD similarity. In (Siolas & d'Alch Buc, 2000) weighting schemes and the RBF kernel were used along with the proximitry matrix. Probably, this combination has downgraded the role of WN semantics. Finally, the experiments were carried out by using only 200 features (selected via Mutual Information statistics). In this way the contribution of rare or non statistically significant terms is neglected. In our view, the latter features may give, instead, a relevant contribution once we move in the SK space generated by the WN similarities.

Other important work on semantic kernel for retrieval has been developed in (Cristianini et al., 2002; Kandola et al., 2002). Two methods for inferring semantic similarity from a corpus were proposed. In the first a system of equations were derived from the dual relation between word-similarity based on document-similarity and viceversa. The equilibrium point was used to derive the semantic similarity measure. The second method models semantic relations by means of a diffusion process on a graph defined by lexicon and co-occurrence information. The major difference with our approach is the use of a different source of prior knowledge, i.e. WN. Similar techniques were also applied in (Hofmann, 2000) to derive a Fisher kernel based on a latent class decomposition of the term-document matrix.

6. Conclusions

The introduction of semantic prior knowledge in IR and TC is important as a way to lower the training set size and thus increase the applicability of Web learning from suitably selected examples. In this paper, we used the conceptual density function on the WordNet (WN) hierarchy to define a document similarity metric and derive a semantic kernel to train Support Vector Machine classifiers. Cross-validation experiments over 8 categories of 20NewsGroups and Reuters over multiple samples have shown that in poor training data conditions, the WN prior knowledge can be effectively used to improve (up to 4.5 absolute percent points, i.e. 10%) the TC accuracy.

These promising results enable a number of future researches: (1) larger scale experiments with different measures and semantic similarity models (e.g. (Resnik, 1997)); (2) domain-driven specialization of the term similarity by selectively tuning WordNet to the target categories, (3) the impact of feature selection on SK, and (4) the extension of the semantic similarity by a general (i.e. non binary) application of the conceptual density model, e.g. the most important category terms as a prior bias for the similarity score.

Acknowledgments

This research is partially supported by the European project, PrestoSpace (FP6-IST-507336).

References

Agirre, E., & Rigau, G. (1996). Word sense disambiguation using conceptual density. *Proceedings of COLING'96, pages 16–22, Copenhagen, Danmark.*.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2001). On feature distributional clustering for text categorization. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 146–153). New Orleans, Louisiana, United States: ACM Press.

Clark, S., & Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Comput. Linguist.*, *28*, 187–206.

Cristianini, N., Shawe-Taylor, J., & Lodhi, H. (2002). Latent semantic kernels. *J. Intell. Inf. Syst.*, *18*, 127–152.

Fellbaum, C. (1998). *Wordnet: An electronic lexical database*. MIT Press.

Haussler, D. (1999). *Convolution kernels on discrete structures*Technical Report UCS-CRL-99-10). University of California Santa Cruz.

Hofmann, T. (2000). Learning probabilistic models of the web. *Research and Development in Information Retrieval* (pp. 369–371).

Joachims, T. (1999). Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*.

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002). Learning semantic similarity. *in Neural Information Processing Systems (NIPS 15) - MIT Press.*.

Kontostathis, A., & Pottenger, W. (2002). Improving retrieval performance with positive and negative equivalence classes of terms.

Li, H., & Abe, N. (1998). Generalizing case frames using a thesaurus and the mdl principle. *Computational Linguistics*, *23*.

Moschitti, A., & Basili, R. (2004). Complex linguistic features for text classification: a comprehensive study. *Proceedings of ECIR-04, 26th European Conference on Information Retrieval*. Sunderland, UK: Springer Verlag.

Resnik, P. (1997). Selectional preference and sense disambiguation. *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?, Washington, April 4-5, 1997.*.

Scott, S., & Matwin, S. (1999). Feature engineering for text classification. *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 379–388). Bled, SL: Morgan Kaufmann Publishers, San Francisco, US.

Siolas, G., & d'Alch Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5* (p. 5205). IEEE Computer Society.

Smeaton, A. F. (1999). Using NLP or NLP resources for information retrieval tasks. In T. Strzalkowski (Ed.), *Natural language information retrieval*, 99–111. Dordrecht, NL: Kluwer Academic Publishers.

Sussua, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. *The Second International Conference on Information and Knowledge Management (CKIM 93)* (pp. 67–74).

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

Voorhees, E. M. (1993). Using wordnet to disambiguate word senses for text retrieval. *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993* (pp. 171–180). ACM.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)* (pp. 61–69). ACM/Springer.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*.