A Semantic Kernel to exploit Linguistic Knowledge

Roberto Basili, Marco Cammisa and Alessandro Moschitti

University of Rome "Tor Vergata", Computer Science Department
00133 Roma (Italy),
{basili,cammisa,moschitti}@info.uniroma2.it

Abstract. Improving accuracy in Information Retrieval tasks via semantic information is a complex problem characterized by three main aspects: the *document representation* model, the *similarity estimation* metric and the *inductive algorithm*. In this paper an original kernel function sensitive to external semantic knowledge is defined as a document similarity model. This *semantic* kernel was tested over a text categorization task, under critical learning conditions (i.e. poor training data). The results of cross-validation experiments suggest that the proposed kernel function can be used as a general model of document similarity for IR tasks.

1 Introduction

Machine learning approaches to specific Information Retrieval (IR) tasks (e.g. ad-hoc document retrieval, text classification, text clustering or question answering) are characterized by three main design choices: a *document representation* model, a *similarity estimation* metric and an *inductive algorithm* that derives the decision function (e.g. category membership).

First, the *feature-based representations* are modeled as *bag of words* made by the terms (or *lexical tokens*) as they appear in the source documents. Data sparseness usually affects this representation as no matching is possible when semantically related but not identical tokens are used. This is particularly true when only small training data sets are available. Attempts to overcome this limitation have inspired several research lines that try to extend the *bag of word* representation with more expressive information: term clusters [1], query expansion via statistical corpus-driven (e.g. [2]) or thesauri-driven term expansion (e.g. [3]).

Second, vector space models make use of *document similarity metrics* [4] between information units (e.g. documents and queries) by mapping the latter in feature vectors whose components are the weights associated with features. The cosine similarity between normalized vectors is the most widely adopted model.

Third, *machine learning* (ML) *algorithms*, e.g. K-Nearest Neighbor [5] and Support Vector Machines [6], are widely used in supervised settings for text categorization, filtering or in user-relevance feedback.

Any ML model for IR should thus benefit from *all* the three design choices. Promising methods for one of the above choices are not sufficient alone to improve the overall accuracy, when not adequately modeled along the other two problem dimensions. For example, semantic information about lexical tokens is expected to improve the similarity estimation as wider (e.g. by using term clusters) or more precise (e.g. by using sense disambiguated words) feature matching is supported. However, every large scale evaluation in IR (ad-hoc retrieval, e.g. [7], or text categorization, e.g. [8]) has shown that this extended information provides poor or no benefit at all (see [9] for a more recent and extensive investigation).

The main problem of term cluster based representations seems the unclear nature of the relationship between the word level and the cluster level matching. Even if (semantic) clusters tend to improve the Recall, lexical tokens are, on a large scale, more accurate (e.g. [10]). When term clusters and simple words are combined, the mixed statistical distributions of individual tokens and sets may be inconsistent with the original ones.

In [3, 11] term clusters are obtained through the synonymy information derivable from WordNet [12]. The empirical evidence is that the misleading information due to the choice of wrong (local) senses causes the overall accuracy to decrease. Word sense disambiguation (WSD) was thus applied beforehand by indexing document by means of disambiguated senses, i.e. synset codes [11, 13, 3, 14, 10]. However, even the state-of-art methods for WSD do not improve accuracy because of the inherent noise introduced by disambiguation mistakes.

Sense disambiguated corpora have been also used to study the relationship between sense and topical information (e.g. IRSemcor, [15]). These benchmarks suggest that there is no systematic correlation between semantic phenomena (e.g. regular polisemy) and topical relatedness, as different domains (or queries) are sensitive to different forms of semantic similarity. Word semantic similarity cannot be directly adopted as a general criteria for computing document (i.e. topical) similarity. Again, extended document representations are more dangerous than beneficial if it is not adequately modeled in the resulting metric space. The semantic expansion of features seems to require a corresponding careful adaptation of both the document similarity model and the adopted learning paradigm.

In this paper, a model for document similarity based on the similarity among words in WN is defined and its application to a supervised text classification task is used for empirical assessment. The WN based word similarity provides semantic expansions of lexical tokens traditionally used as features for a document (Section 2). A corresponding novel vector space model is then proposed where features are pairs of similar words (Section 3). Intuitively, every document d is represented through the set of all pairs $< t, t' >$ originating by terms $t \in d$ and some words t' *enough similar* to t. In this space the same pairs found in different documents contribute to their similarity, even if originating tokens are different. No sense is *a priori* pruned from a document representation but sense matching is triggered only when document matching is carried out.

Such space may be composed by $O(|V|^2)$ dimensions, where V is the corpus vocabulary. If we consider only the WN nouns, the space may contain about 10^{10} features. This critical complexity impacts on the learning algorithm. However, kernel methods can be applied as they can represent feature spaces implicitly. Among kernel-based learners, Support Vector Machines (SVMs) [16] have shown to achieve high accuracy by dealing effectively with many irrelevant features. Here, the selection of the suitable pairs is left to the SVM learning. Therefore, no sense disambiguation is imposed *a priori*, but sense selection is carried out *on the fly*. The overall model is thus distinct from most of the previous work in language-oriented IR.

The improvements in the overall accuracy observed over a TC task (Section 4) make this model a promising document similarity model for general IR tasks: unlike previous attempts, it makes sense of the adoption of semantic external resources (i.e. WN) in IR.

2 A semantic similarity measure

Semantic generalizations overcome data sparseness problems in IR as contributions from different but semantically similar words are still available.

Methods for corpus-driven induction of semantically inspired word classes have been widely used on language modeling and lexical acquisition tasks (e.g. [17]). The main resource employed in most works is WordNet [12]. The WordNet noun hierarchy represents lexicalized concepts (or senses) organized according to an "*is-a-kind-of*" relation. A concept s, labeled with words w used to denote it, is thus called a *synset*, $syn(s)$. Words w are synonyms under the specific dimension captured by s and every synset s is a (lexical) sense for all the members $w \in syn(s)$.

The noun hierarchy is a direct acyclic graph[1]. The *direct_isa* relation defined by edges in the graph can be extended via a transitive closure to determine the overall *isa* relation between pairs of synsets. In line with [17] we denote by $\bar{s}$ the set of nodes in the hierarchy dominated by s, i.e. $\{c|c \ isa \ s\}$. By definition $\forall s \in \bar{s}$.

The automatic usage of WordNet for NLP and IR tasks has proved to be very complex. First, how the topological distance between 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. Moreover, the approximation of concepts by means of their shared generalizations in the hierarchy implies a conceptual loss that impacts on the target IR (or NLP) tasks. Similar words play different roles in IR tasks, so that equivalence cannot be imposed in general. This depends on the lack of semantic properties needed to select the word topical roles. It is thus difficult to decide the degree of generalization at which the conflation of senses into single features can be effective for IR. Attempts to automatically determine suitable levels (as

[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 exceptions heuristically.

'cuts' in the hierarchy) has been proposed in [18] with justifications derived from corpus statistics. For several tasks (e.g. in TC) this is unsatisfactory: different contexts (e.g. documents) may require different generalizations of the same word as they independently impact on the suitable document similarity. This is one of the limitations of corpus-driven metrics, like the one proposed by [19].

A flexible notion of semantic similarity is the *Conceptual Density* (CD) measure, early introduced in [20]. It depends on the generalizations of word senses not referring to any fixed level of the hierarchy. The measure used in this paper corresponds to the CD variant defined in [21], applied to semantic tagging and lexical alignment for ontology engineering. CD defines a distance between lexicalized concepts according to the topological structure of WordNet and can be seemingly applied to two or more words.

Conceptual Density (CD) makes a guess about the proximity of senses, s_1 and s_2, of two words u_1 and u_2, according to the information expressed by the (minimal, i.e. maximally dense) subhierarchy, $\bar{s}$, that includes them. Let S_i be the set of generalizations for at least one sense s_i for the word u_i, i.e. $\{s|s_i \in \bar{s}\}$. Given two words u_1 and u_2, their CD is formally defined as:

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

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 actual sub-hierarchy $\bar{s}$. $\mu(\bar{s})$ depends on WordNet and in some cases its value can approach 1.
- $|\bar{s}|$ is the number of nodes in the sub-hierarchy $\bar{s}$. This value is statically estimated from WN and it is a negative bias for higher level generalizations (i.e. larger $\bar{s}$).
- h is the depth of the *ideal* WN subtree able to represent the lexical senses s_1 and s_2 of the two words. This value is actually estimated by:

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

  where h expresses, given the average branching factor $\mu(\bar{s})$ at $\bar{s}$, the minimal number of levels needed to have s_1, s_2 represented in the leaves. Eq. 2 prevents the logarithm to assume an infinite value in cases $\mu(s)$ is exactly 1.

Conceptual density models the semantic distance as the density of the most dense generalization $\bar{s}$ such that $s \in S_1 \cap S_2$. The *density* of $\bar{s}$, is the ratio between the number of its useful nodes and $|\bar{s}|$. Useful nodes are those referring to senses of the involved words, i.e. s_1 and s_2. The density accounts for the branching factor local to $\bar{s}$: the higher is $\mu(\bar{s})$, the lower is the hierarchy height (h) sufficient to represent lexical senses (s_1 and s_2) with the highest density. If u_1 and u_2 are synonyms, the similarity measure gives 1, i.e. the highest similarity. Notice 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. It must be noticed that Eq. 1 is the binary version of the general model defined in [21].

3 A WordNet Kernel for document similarity

Term similarity is used in the design of the document similarity which is the core function of most learning algorithms for TC. Document similarity models based on string matching do not support functions much different from the (inner) products between weights (of matching terms). The term similarity proposed in Eq. 1 is defined 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.

In order to model all pair contributions, we will still define a document similarity function as an inner product but in a new vector space where, intuitively, the dimensions are all possible pairs in the initial vocabulary and the weights of such components depend on the term similarity function. 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[2] is enough to show that it is a specialization of the general definition of convolution kernels formalized in [23]. Hereafter, we report such definition: let $X, X_1, .., X_m$ be separable metric spaces, $x \in X$ a structure and $\boldsymbol{x} = x_1, ..., x_m$ 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 $R(\boldsymbol{x}, x)$ is "true" if $\boldsymbol{x}$ are the parts of x. We indicate with $R^{-1}(x)$ the set $\{\boldsymbol{x} : R(\boldsymbol{x}, x)\}$. Given two objects x and $y \in X$ their similarity $K(x, y)$ is defined as:

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

[2] An alternative way to prove the validity of the Mercer's conditions was shown in [22]. It is enough to observe that the kernel $K(d_1, d_2)$ can be written as $\boldsymbol{\lambda}_1 W \cdot W' \boldsymbol{\lambda}_2$, where $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are the vectors of weights associated with d_1 and d_2, and W and W' are the matrix and its transposed of the WordNet term similarities. Clearly, $P = W \cdot W'$ is positive semi-definite, thus $K(d_1, d_2) = \boldsymbol{\lambda}_1 P \boldsymbol{\lambda}_2$ satisfies the Mercer's conditions. Note that this proof does not show that our kernel is a convolution kernel.

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), $\boldsymbol{x} = x_1 = w \in V$ (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 the learning algorithm of 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, $H(\boldsymbol{x}) = \boldsymbol{\omega} \cdot \boldsymbol{x} + b = 0$, where $\boldsymbol{x}$ and $\boldsymbol{\omega} \in \mathbb{R}^\eta$ and $b \in \mathbb{R}$ are learned by applying the *Structural Risk Minimization principle* [16]. From the kernel theory we have that:

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

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

where, d is a classifying document and d_h are all the l training instances, projected in $\boldsymbol{x}$ and $\boldsymbol{x}_h$ 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 $\boldsymbol{x_h}$ or $\boldsymbol{x}$. 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 of the Eq. 3 is that we can consider only the word pairs associated with non-zero weight, 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:

$$SK'(d_1, d_2) = \frac{SK(d_1, d_2)}{\sqrt{SK(d_1, d_1) \cdot SK(d_2, d_2)}} \tag{6}$$

It should be noted that, the sparse evaluation also has a quadratic time complexity which is much less efficient than the linear complexity of the traditional document similarity. This, prevents the use of large document sets in the experiments. Moreover, as we claim that the general prior knowledge provided by WordNet can be effective only in poor training data conditions, we carried out cross-validation experiments on small subsets of the well known TC corpus 20 NewsGroups (20NG). It is available at `www.ai.mit.edu/ people/jrennie/20Newsgroups/` and contains a general terminology which is mostly covered by in WN.

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 assessed experimentally. 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 is due to the large dimension of the similarity matrix, i.e. in principle any pair of WN words have a non null similarity score. However, it has to be evaluated only once. Moreover, we are not interested to large collections of training documents as simple *bag-of-words* models are in general very effective [9], i.e. they seems to model well the document similarity needed by the learning algorithm. For any test document, in fact, a set of support vectors can be found able to suggest similarity according to a simple string matching model. In other words, training documents are available including a large number of terms found in the target test document. We selected small subsets from the 20NewGroups collection, instead, and, in order to simulate critical learning conditions, experiments were run on training sets of increasing size.

4.1 Experimental set-up

In order to get statistically significant results, 10 different samples of 40 documents were randomly extracted, from 8 out of the 20 categories of the Usenet newsgroups collection. The training was carried out over the 10 distinct samples. For each learning phase, one sample was used as a validation set and the remaining 8 as test-set. This means that we run 80 different experiments for each model.

The classifier runs were carried out by using the SVM-light software [6] (available at `svmlight.joachims.org`) with the default linear kernel on the token space adopted as the baseline evaluations. The semantic kernel SK was implemented inside SVM-light.

The SK kernel (in Eq. 3) was applied with $\sigma(\cdot,\cdot) = CD(\cdot,\cdot)$ (Eq. 1), i.e. it is sensitive only to noun information. Accordingly, part of speech tagging was applied. However, verbs, adjectives and numerical features were used in all the experiments: in the space of lexical pairs, they have a null similarity with respect to any other word.

The classification performances were evaluated using the f_1 measure[3] for single arguments and the MicroAverage for the final classifier pool [24]. The performance are expressed as the mean and the standard deviation over 80 evaluations.

Given the small number of documents careful SVM parameterization was applied. Preliminary investigation suggested that the trade-off (between the training-set error and margin) parameter, (i.e. c option in SVM-light) 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

[3] f_1 assigns equal importance to Precision P and Recall R, i.e. $f_1 = \frac{2P \cdot R}{P+R}$.

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

the accuracy. Finally, feature selection techniques and weighting schemes were not applied in our experiments, as they cannot be accurately estimated from the small training data available.

4.2 Cross validation results

The SK (Eq. 3) was compared with the linear kernel which obtained the best f_1 measure in [6]. Table 1 reports the first comparative results for three categories (about 15 training documents each). Global results were obtained by averaging over 80 runs of the same size. The *Mean* and the *Std. Dev.* of f_1 are reported in Column 2 (for linear kernel SVMs), Column 3 (SK as in Eq. 3 without applying POS information, i.e. no noun selection applied) and Column 4 (SK with the use of POS information). The last row shows the Microaverage performance for the above three models.

Category	*Bow*	SK	SK-POS
Atheism	59.6±11.2	63.7±10.7	63.0±9.6
Talk.Relig.	63.5±10.6	66.0±7.8	64.9±8.5
Comp.Graph.	85.3±8.3	86.7±7.4	85.7±9.8
MicroAvg. f_1	68.6±5.0	72.2±5.4	71.4±5.5

Table 1. SVM performance using the linear and the Semantic Kernel over 3 categories of 20NewsGroups with 40 documents of training data.

In order to asses these findings we repeated the evaluation over 8 20NewGroups categories (about 5 documents each). The results are reported in Table 2.

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

All the results confirm that SK outperforms the best *bow* linear kernel of about 4% as in critical learning conditions the semantic contribution of the SK recovers with useful information. In particular, during the similarity estimation, a word in a document activates 60.05 pairs (i.e. the other words in the matching document), on average. This is particularly useful to increase the amount of information available to the SVM. Noise due to semantic ambiguity seems not harmful to the SVM learner.

First, only the useful information seems to be made available by the training examples: similar words according to the correct senses appear in the positive examples so that only the useful pairs are amplified. Moreover, noisy information seems to be tolerated by the robustness of the learning SVM algorithm. Irrelevant pairs (senses emerging by mistakes) have a smoother distribution and are neglected by the SVM algorithm.

Second, the Standard Deviations tend to assume high values. However, given the high number of samples the results are statistically reliable. To verify such hypothesis, we carried out the Normal Distribution confidence test on the 80 samples. The stochastic variable observed in each sample was the differences between the Microaverage of SK and *bow* models. The result shows that SK reaches higher Microaverage than the baseline at 99% of confidence level.

Third, a study on the impact of training data set size on the learning accuracy has been carried out for the three above models: *bag-of-words* (*bow*), SK and *SK-POS*. Figure 1 shows the derived leaning curves over training data set of increasing size (5,10 and 15 documents for each category).

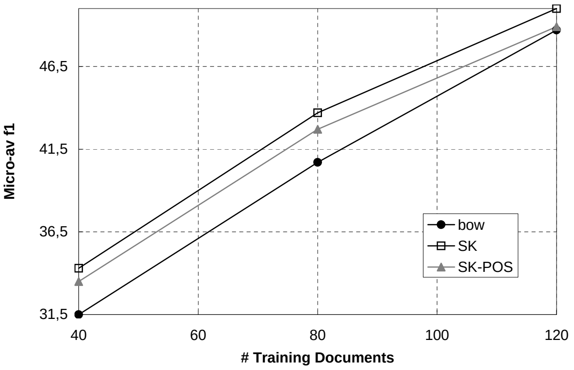

Fig. 1. MicroAverage f_1 of SVMs using *bow*, SK and SK-POS kernels over the 8 categories of 20NewsGroups.

As expected the advantage of SK tends to reduce when more training data is available. However, the improvements keep not negligible. The SK model (without POS information) still preserves about 3% improvement. The similarity matching possibly allowed between noun-verb pairs still captures semantic information useful for topical similarity.

Finally, an experiment with 3 categories (compare with Table 1) was made by discarding all string matchings from SK. Only words having different surface forms were allowed to give contributions to Eq. 3. An important outcome is that the SK converges to an f_1 value of 50.2%. This shows that the word similarity provided by WN is consistent and effective to TC.

5 Related Work

The IR work related to this study focus on similarity (clustering) models for embedding statistical and external knowledge in document similarity.

In [25] a *Latent Semantic Indexing* analysis was used for term clustering. The algorithm, as described in [26], assumes that values x_{ij} in the transformed term-term matrix represents the similarity (> 0) and anti-similarity between terms i and j. 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.

In [1], a feature selection technique that clusters similar features/words, called the Information Bottleneck (IB), is applied to TC. Support Vector Machines trained over clusters were experimented on three different corpora: *Reuters-21578*, WebKB and 20NewsGroups. Controversial results are obtained as the cluster based representation outperformed the simple *bag-of-words* only on the latter collection (>3%).

The use of external semantic knowledge seems to be more problematic in IR as the negative impact of semantic ambiguity [11]. 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 [13] and [3, 14]. Word semantic information was here used for text indexing and query expansion, respectively. In [14] it is shown that semantic information derived directly from WN without a priori WSD produces poor results.

The above methods are even more problematic in TC [9]. 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 seems to suffice for automatic disambiguation. [27] uses 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 [10] (3 corpora using cross-validation techniques) showed that even accurate disambiguation of WN senses (about 80% accuracy on nouns) does not improve TC.

An approach similar to the one proposed in this article, is the use of term proximity to define a semantic kernel [28]. Such semantic kernel was designed as a combination of the Radial Basis Function 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 difference with our approach are the following: 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. [19]) and should be always applied. Second, the description of the resulting space is not given and the choice of the kernel is not justified in terms of document similarity. The proximity matrix is a way to smooth the similarity between two terms but its impact on learning is unclear. Finally, 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 work using corpus statistc knowledge, e.g. latent semantic indexing, for retrieval was carried out in [29, 30].

6 Conclusions

The introduction of semantic prior knowledge in IR has always been an interesting subject as the examined literature suggests. In this paper, we used the conceptual density function on the WordNet hierarchy to define a document similarity metric. Accordingly, we defined a semantic kernel to train a Support Vector Machine classifiers. Cross-validation experiments over 8 categories of 20NewsGroups over multiple samples have shown that in critical training data conditions, such prior knowledge can be effectively used to improve (about 3 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. [19]); (2) domain-driven specialization of the term similarity, by selectively tuning WordNet to the target categories, (3) optimization driven by prior feature selection, and (4) extension of the semantic similarity by a general (i.e. non binary) application of the conceptual density model, e.g. use the most important category terms as prior bias for the similarity score.

References

1. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: On feature distributional clustering for text categorization. In proceedings of SIGIR'01, New Orleans, Louisiana, United States, ACM Press (2001)
2. Strzalkowski, T., Carballo, J.P.: Natural language information retrieval: TREC-6 report. In: Text REtrieval Conference. (1997)
3. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In proceedings of SIGIR'93. Pittsburgh, PA, USA, 1993
4. Salton, G.: Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley (1989)
5. Yang, Y.: Expert network: effective and efficient learning from human decisions in text categorisation and retrieval. In proceedings of SIGIR'94, Dublin, IE (1994)
6. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning. (1999)
7. Strzalkowski, T., Carballo, J.P., Karlgren, J., Tapanainen, A.H.P., Jarvinen, T.: Natural language information retrieval: TREC-8 report. In: Text REtrieval Conference. (1999)

8. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In proceedings of SIGIR'92, Kobenhavn, DK (1992) 37–50
9. Moschitti, A.: Natural Language Processing and Automated Text Categorization: a study on the reciprocal beneficial interactions. PhD thesis, Computer Science Department, Univ. of Rome "Tor Vergata" (2003)
10. Moschitti, A., Basili, R.: Complex linguistic features for text classification: a comprehensive study. In proceedings of ECIR'04, Sunderland, UK, Springer Verlag (2004)
11. Smeaton, A.F.: Using NLP or NLP resources for information retrieval tasks. In Strzalkowski, T., ed.: Natural language information retrieval. Kluwer Academic Publishers, Dordrecht, NL (1999)
12. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press. (1998)
13. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In proceedings of CKIM'93. (1993)
14. Voorhees, E.M.: Query expansion using lexical-semantic relations. In proceedings of SIGIR'94, Dublin, Ireland, (1994)
15. Fernandez-Amoros, D., Gonzalo, J., Verdejo, F.: The role of conceptual relations in word sense disambiguation. In proceedings of the 6th international workshop on applications of Natural Language for Information Systems (NLDB 2001). (2001)
16. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
17. Clark, S., Weir, D.: Class-based probability estimation using a semantic hierarchy. Computional Linguistics (2002)
18. Li, H., Abe, N.: Generalizing case frames using a thesaurus and the mdl principle. Computational Linguistics (1998)
19. Resnik, P.: Selectional preference and sense disambiguation. In proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?, Washington, April 4-5, 1997. (1997)
20. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. In proceedings of COLING'96, pages 16–22, Copenhagen, Danmark. (1996)
21. Basili, R., Cammisa, M., Zanzotto, F.M.: A similarity measure for unsupervised semantic disambiguation. In proceedings of Language Resources and Evaluation Conference. (2004)
22. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines. Cambridge University Press (2000)
23. Haussler, D.: Convolution kernels on discrete structures. Technical report ucs-crl-99-10, University of California Santa Cruz (1999)
24. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval Journal (1999)
25. Kontostathis, A., Pottenger, W.: Improving retrieval performance with positive and negative equivalence classes of terms (2002)
26. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science (1990)
27. Scott, S., Matwin, S.: Feature engineering for text classification. In Bratko, I., Dzeroski, S., eds.: Proceedings of ICML'99, San Francisco, US (1999)
28. Siolas, G., d'Alch Buc, F.: Support vector machines based on a semantic kernel for text categorization. In proceedings of IJCNN'00, IEEE Computer Society (2000)
29. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. J. Intell. Inf. Syst. (2002)
30. Kandola, J., Shawe-Taylor, J., Cristianini, N.: Learning semantic similarity. In NIPS'02 - MIT Press. (2002)