Semantic Kernels for Text Classification based on Topological Measures of Feature Similarity

Stephan Bloehdorn
Institute AIFB, Knowledge Management Group
University of Karlsruhe, Germany
bloehdorn@aifb.uni-karlsruhe.de

Roberto Basili, Marco Cammisa and
Alessandro Moschitti
University of Rome 'Tor Vergata', Italy
{basili,cammisa,moschitti}@info.uniroma2.it

Abstract

In this paper we propose a new approach to the design of semantic smoothing kernels for text classification. These kernels implicitly encode a superconcept expansion in a semantic network using well-known measures of term similarity. The experimental evaluation on two different datasets indicates that our approach consistently improves performance in situations of little training data and data sparseness.

1 Introduction

Text classification systems which automatically organize documents into predefined classes are one approach to govern the complexity of handling the ever growing amounts of textual data. Most systems employ machine learning methods among which Support Vector Machines (SVMs) along with other kernel-based algorithms have become a dominant technique. The popularity of SVMs stems from (i) a firm grounding in statistical learning theory, (ii) high generalization capabilities, and (iii) their ability to incorporate prior knowledge about the target domain by means of a specific choice of the employed *kernel function* [7]. Most approaches for text classification with SVMs have exploited the standard *bag-of-words* feature representation originating from Information Retrieval. Here, documents are encoded as vectors with dimensions corresponding to corpus terms and vector components corresponding to (weighted) counts of the respective terms in the document and the plain inner product or the cosine between two vectors are used as kernel. Obviously, the index terms that constitute the feature space can not be regarded as mutually orthogonal dimensions but rather as dimensions with varying degrees of semantic similarity. However, studies indicate that the *bag-of-words* approach achieves very good results in cases where sufficient training data is available as reliable patterns can be detected in these cases. In contrast, in cases where training data is scarce or the representation of individual instances is hampered by extreme sparseness, an a-priori bias in form of a more adequate kernel is likely to boost the overall performance. *Semantic smoothing kernels* that encode semantic dependencies between terms have emerged as one paradigm to approach this problem [8, 6, 1]. In this paper, we investigate the use of a new type of semantic smoothing kernels for text classification. Our smoothing kernel implicitely represents the index terms as instances within a seperate *term space* through vectorial representations of superconcept expansions. We use a variety of measures of lexical semantic relatedness (cf section 2) for weighted representations in this term space embedding. The mutual term similarities are calculated as inner products in this term space. For calculating document similarities, these are used in a generalized dot product where also vector components *across dimensions* contribute to the kernel evaluation of two documents (cf section 3). Results in two series of experiments that mirror typical situations of little training data and data sparseness – subsets of Reuters-21578 and the TREC question classification corpus – indicate a consistent improvement of classifier performance using our smoothing kernel (cf section 4).

2 Preliminaries

Semantic Networks The target semantic dependencies are encoded in structures which we call, for simplicity, *semantic networks* which can be seen as directed graphs connecting concept nodes by directed links. For two concepts c_1, c_2 the relation $super(c_1, c_2)$ indicates that c_2 is *superconcept of* c_1. In this work, we restrict our attention to

Acknowledgements: This research was partially supported by the European Commission under contracts IST-2003-506826 SEKT and IST-2005-026978 XMedia. The expressed content is the view of the author(s) but not necessarily the view of the project consortia as wholes.

WordNet[1], a free lexical reference system and semantic network that organizes English terms into interconnected groups of synonyms (*synsets*). The measures introduced next require three further notions. By *distance* (d) of two concepts c_1 and c_2, we refer to the number of superconcept edges between c_1 and c_2 which can be easily computed using the Floyd-Warshall algorithm. The notion of the *depth* (*dep*) of a concept relates to the frequent assumption of a tree-like structure of the semantic network having a unique root element. For an acyclic graph (which we assume in the remainder), a root element can be introduced which becomes superconcept of all concept nodes that are not equipped with outgoing superconcept edges. The depth of a concept is then defined as the distance of the concept to the root. Based on this, the *lowest super ordinate (lso)* of two concepts refers to the concept with maximal depth that subsumes them both.

Measuring Semantic Relatedness The measurement of semantic similarity is a problem that pervades Ccomputational Linguistics with respect to a large number of applications in Natural Language Processing. We give a brief review of the measures used later on and point the interested reader to [3] for a detailed and recent survey.

The *inverted path length* is the simplest way for computing the semantic similarity between two concepts:

$$sim_{IPL}(c_1, c_2) = \frac{1}{(1 + d(c_1, c_2))^{\alpha}},$$

whereby α specifies the rate of decay. Note that [8] have used this measure to define semantic smoothing kernels for the first time. Despite being simple and intuitive, the inverted path length does not comply with the intuition that concepts closer to the root of the semantic network should have a higher distance compared to concepts far away. Among others, the *Wu&Palmer measure* tries to scale the similarity with respect to the depth of the concepts and their lowest super ordinate:

$$sim_{WUP}(c_1, c_2) = \frac{2\,dep(lso(c_1, c_2))}{d(c_1, lso(c_1, c_2)) + d(c_2, lso(c_1, c_2)) + 2\,dep(lso(c_1, c_2))}.$$

A different type of measure incorporates knowledge about the *information content* of a concept besides the structural setup of the semantic network. The intuation behind the *Resnik Measure* is that neither the individual edges nor the absolute depth in a taxonomy can be considered as homogeneous indicators of the semantic content of a concept. Instead, concept similarity is measured as follows[2]:

$$sim_{RES}(c_1, c_2) = -\log\ P(lso(c_1, c_2)).$$

As an extension to the Resnik Measure, the *Lin Measure* uses the information content of the compared concepts is used as a means for normalization:

$$sim_{LIN}(c_1, c_2) = \frac{2\ \log\ P(lso(c_1, c_2))}{\log\ P(c_1) + \log P\,(c_2)}.$$

3 Designing Semantic Kernels

As motivated in section 1, the aim of our work is to embed the knowledge about the topological relations of the semantic networks in kernel functions. This allows the learning algorithm to relate distinct but similar features during kernel evaluation.

Semantic Kernels The general concept of semantic smoothing kernels was for the first time introduced in [8] and subsequently revisited in [4, 6, 1], each time based on different design principles.

Definition 1 (Kernel Function)**.** *Any function κ that for all $x, z \in X$ satisfies $\kappa(x, z) = \langle\phi(x), \phi(z)\rangle$, is a valid kernel, whereby X is the input vector space under consideration and ϕ is a suitable mapping from X to a feature space F.*

Definition 2 (Semantic Smoothing Kernel)**.** *The semantic smoothing kernel for two data items (documents) $x, z \in X$ is given by $\kappa(x, z) = x'Qz$ where Q is a square symmetric matrix whose entries represent the semantic proximity between the dimensions of the input space X.*

Note that the definition of a kernel implies that Q must be a positive semi-definite matrix. Conceptually this means that Q can be decomposed by $Q = PP'$ thus revealing the underlying feature mapping as $\phi(x) = P'x$. The matrix P is a $n \times m$ matrix whereby n corresponds to the dimensionality of the input space X and provides a linear transformation of the input document into a feature space of (possibly far higher) dimensionality m, similar to a query expansion. A first approach to designing semantic kernels would be to embed the pairwise measures of lexical semantic relatedness directly into the matrix Q. However, the requirement of Q being positive semi-definite can typically not be ensured in general if used directly.

[1] http://www.cogsci.princeton.edu/~wn/

[2] The measure uses the notion of the probability $P(c)$ of encountering a concept c which can be estimated from corpus statistics. It follows the trail of information theory in quantifying the *information concept (IC)* of an observation as the negative log likelihood. Intuitively, a universal root concept having a probability of 1 carries an information content equal to zero while rare concepts carry high information content values.

Semantic Kernels based on Superconcept Expansions As a way to avoid indefinite similarity matrices, authors like [8] have enforced the positive-definiteness of Q by exlicitely computing it from $Q = PP'$ whereby the information about the similarities is now encoded in the matrix P. While this approach ensures the validity of the Kernel, the interpretation of the resulting smoothing kernel is less clear. Conceptually, it maps each concept to a number of related concepts and the shared weight of these determines the overall similarity between two terms.

Following own prior work in a differnet setting [2], we follow a different approach for the construction of Q which is, however, also based on an explicit construction of the type $Q = PP'$. We choose a setup of P such that it provides a mapping into the space of all possible *superconcepts* of the input instances, i.e. the terms or concepts in question. That is, the rows of P correspond to vector representations of the concepts of the input space by means of their respective superconcepts. The similarity of two concepts in the resulting smoothing matrix Q is thus the dot product of the vectors of their respective superconcepts. This approach is intuitive as we can typically regard two concepts as similar if they share a large number of superconcepts as opposed to sharing only few superconcepts.

Recently, [6] have investigated this approach motivated by the observation that the dot product of two terms represented as vectors of their respective superconcepts can be shown to be equivalent to a number of popular similarity measures (among them the Resnik measure, but not the Lin and Wup measures) given a particular weighting scheme of the superconcept representation. However, this prior work has focused on the simple case of giving the superconcepts in the mapping P full and equal weight (i.e. restricting P to a 0/1 matrix) while varying the number of superconcepts that are considered. Consistent with an argument made by the same authors, we argue that the variation of the number of superconcepts yields a high variance and its a-priori choice will always be an ad-hoc decision.

As an alternative approach, we have investigated the use of different weighting schemes for the representation of the superconcepts in P motivated by the following rationales: (i) the weight a superconcept c_j receives in the vectorial description of a concept c_i should be influenced by its distance from c_i and (ii) the weight a superconcept c_j receives in the vectorial description of a concept c_i should be influenced by its overall depth in the semantic network.

Based on these rationales and the measures introduced in section 2, we have investigated the following weighting schemes:

full: No weighting, i.e. $P_{ij} = 1$ for all superconcepts c_j of c_i and $P_{ij} = 0$ otherwise.

full-ic: Weighting using information content of c_j, i.e. $P_{ij} = sim_{RES}(c_i, c_j)$ and $P_{ij} = 0$ otherwise.

path-1: Weighting based on inverted path length, i.e. $P_{ij} = sim_{IPL}(c_i, c_j)$ for all superconcepts c_j of c_i and $P_{ij} = 0$ otherwise using the parameter $\alpha = 1$.

path-2: The same but using the parameter $\alpha = 2$.

lin: Weighting using the Lin similarity measure, i.e. $P_{ij} = sim_{LIN}(c_i, c_j)$ and $P_{ij} = 0$ otherwise.

wup: Weighting using the Wu&Palmer similarity measure, i.e. $P_{ij} = sim_{WUP}(c_i, c_j)$ and $P_{ij} = 0$ otherwise.

The different weighting schemes behave differently wrt the above motivations. While full does not implement any of them, full-ic considers rationale (ii) while path-1 and path-2 consider rationale (i). The schemes lin and wup reflect combinations of both.

4 Experimental Evaluation

Experimental Setup In this section, we report on experiments on the Reuters-21578 and the TREC QA datasets. We implemented the semantic kernel within a custom kernel module for the current version of SVM-light [5] which is freely available for download[3]. In the experiments, we used the noun hierarchy of WordNet as the underlying semantic network. The setup of the smoothing matrices used in the evaluation experiments was based on the particular choice of the proximity matrix design as discussed in section 3, as well as on two simplifying assumptions. Firstly, the existing bag-of-word representation of the documents required the design of a *term proximity matrix* as opposed to the *synset proximity matrix* assumed so far. We used a simple strategy that maps each term to its most frequent noun sense (if it exists). Note that this approach implies an inherent word sense disambiguation side effect. Secondly, in the case of the Reuters-21578 experiments, we restricted the entries in the term proximity matrix to those terms having document frequencies of at least five. Entries that were undefined in the term proximity matrix – be it because a missing mapping to a noun synset or because of low document frequency – were assumed to take the default values (i.e. zero and one for off-diagonal and diagonal entries). Frequency counts needed for the calculation of the measures making use of information content were obtained from (i) the complete Reuters-21578 collection in the case of the Reuters-21578 experiments or (ii) from the Brown corpus in the case of the experiments on the TREC question dataset.

Experiments on Reuters-21578 As basis for our experiments on Reuters-21578 we used the 'ModApte' split which divides the Reuters-21578 collection into 9,603 training

[3] http://www.aifb.uni-karlsruhe.de/WBS/sbl/software/semkernel/

documents, 3,299 test documents and 8,676 unused documents. We prepared the bag-of-words representation of the documents based on the standard preprocessing steps, namely tokenization, removal of the standard stopwords for English defined in the SMART stopword list, lemmatization and TFIDF weighting. For the purpose of quantifying performance gains in cases where very little training is available, we prepared small subsets by randomly choosing 2%, 3%, 4% and 5% of the ModeApte training data[4]. To account for the high inherent sampling variance, this approach was repeated 10 times for each of the 4 subset sizes resulting in a total number of 40 subsets. Binary classification experiments were then conducted for the 10 largest Reuters-21578 categories and each subset, resulting in a total number of 400 experiments in each run. The corresponding testing was conducted using the full test set. The 'soft margin' parameter c that controls the influence of misclassified examples was set to $c = 0.1$ in all experiments.

Table 1 summarizes the absolute macro F_1 values obtained over the different subsets of Reuters-21578 as explained above. The results indicate a consistent improvement of the F_1 values for all of the smoothing kernels based on superconcept representations. The extent of the improvement for the smoothing kernels based on superconcept representations relative to the linear kernel can be seen more clearly in figure 1. In line with the prior findings in [6], the improvement gradually diminishes as more training data becomes available. Among the different weighting schemes for superconcept representations, the lin weighting scheme that combines the 'distance' and 'depth' considerations tends to outperform the other measures. On the contrary, the default scheme (full) that does not employ any weighting schemes tends to be inferior to other models that use them.

	Subset Size			
kernel	**2%**	**3%**	**4%**	**5%**
linear	0.45	0.51	0.54	0.57
full	0.50	0.53	0.57	0.58
full-ic	0.53*	0.55	0.60	0.61
path-1	0.50	0.54	0.59	0.61
path-2	0.48	0.53	0.57	0.59
lin	0.53*	0.57*	0.61*	0.62*
wup	0.52	0.55	0.59	0.61

Table 1. Absolute macro F1 scores for Reuters-21578 subsets.

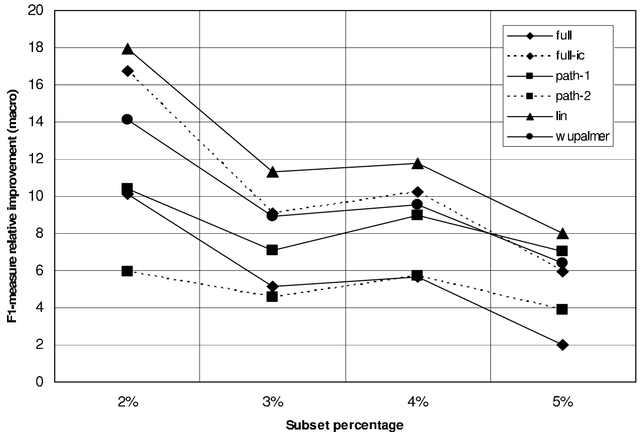

Figure 1. Relative improvements of macro F1 scores for Reuters-21578 subsets.

Experiments on the TREC Question Classification Dataset The long tradition of Question Answering (QA) in TREC has produced a large question set which has recently been used for Question Classification (QC) tasks [9]. Note that, compared to standard text classification settings, questions contain only extremely few words which makes this setting a typical victim of data sparseness. We have used a set of questions, labeled according to the *coarse grained* classification as described in [9]. The dataset is divided into 5,500 questions for training and the 500 questions for testing[5]. We preprocessed the questions in the same way as in the first experiment and again performed binary classification experiments on each of the 6 question types. In this experiment, we additionally applied several values of the 'soft margin' parameter c since our preliminary tests showed that its variation has an important influence on the overall results. Starting from $c = 0.1$ and $c = 1.0$ as typical default choices, we varied these in three steps to $c = 0.1 \ldots 0.3$ and $c = 1 \ldots 3$. Table 2 summarizes the absolute macro F_1 as well as the micro F_1 values obtained in the question classification setting. The best values per setting of c are highlighted.

Results indicate a consistent superior accuracy of the semantic smoothing kernels over the linear kernel baseline. With the exception of the full-ic setup, which shows good results for small values of c but detoriates later on, all semantic smoothing kernels improve performace in both the macro- as well as micro-averaged setting. In line with the results on the Reuters-21578 experiments, the lin scheme

[4] While we were not primarily interested in the application of our approach in those cases where sufficient training data is available, we have nevertheless investigated the effect of superconcept smoothing kernels together with the full ModeApte training set. Results indicate only little shifts in performance, sometimes even degrading performance which supports our assumption that the smoothing is not particularly useful in this scenario.

[5] These training questions are selected from the 4500 English questions published by USC, 500 questions annotated for rare classes and 894 questions from TREC 8 and TREC 9, the test questions are taken from TREC 10. The dataset is freely available at `http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/`. The *coarse grained* classification defines 6 classes: *Abbreviations, Descriptions, Entity, Human, Location, Numeric.*

macro-avg	soft margin parameter *c*					
kernel	**0.1**	**0.2**	**0.3**	**1.0**	**2.0**	**3.0**
linear	0.21	0.38	0.47	0.62	0.63	0.64
full	0.38	0.49	0.55	0.61	0.61	0.68
full-ic	0.53*	0.53*	0.53	0.62	0.55	0.55
path-1	0.25	0.42	0.51	0.64*	0.64	0.64
path-2	0.22	0.39	0.47	0.63	0.65*	0.64
lin	0.36	0.49	0.56*	0.64*	0.62	0.70*
wup	0.34	0.49	0.54	0.62	0.61	0.69

micro-avg	soft margin parameter *c*					
kernel	**0.1**	**0.2**	**0.3**	**1.0**	**2.0**	**3.0**
linear	0.09	0.25	0.34	0.55	0.57	0.58
full	0.27	0.38	0.45	0.55	0.56	0.68
full-ic	0.47*	0.46*	0.47*	0.60*	0.49	0.48
path-1	0.14	0.32	0.40	0.57	0.58	0.59
path-2	0.08	0.28	0.37	0.57	0.59*	0.58
lin	0.27	0.37	0.47*	0.57	0.57	0.69*
wup	0.23	0.37	0.45	0.56	0.56	0.68

Table 2. Absolute macro and micro F1 results for TREC-QC, for different values of c.

achieves the best overall performance with a relative improvement of 9.32% for the macro F_1 value in the case of $c = 3$ (i.e. the setting for which the linear kernel achieves its maximum). We generally note that the improvements are more extreme for the case of small values of c while they appear more stable for larger values.

5 Related Work

Semantic kernels were initially introduced in [8] using inverted path length as a similarity measure and subsequently explored in [1] using conceptual density as a similarity measure among others. An alternative approach reported in [4] aimed at incorporating the well-established technique of Latent Semantic Indexing (LSI) into the semantic kernel paradigm. Recently [6] reported on experiments with semantic smoothing kernels defined on superconcept representations such that it forms a natural basis for our work. In contrast to our approach, the authors used extensive word sense disambiguation (WSD) machinery which also formed a core contribution. Similar to [2], the superconcept representations of terms were built upon fixed numbers of superconcepts without further weighting.

6 Conclusion

In this paper, we have investigated the design of semantic smoothing kernels. By expressing the similarity of term features by means of the shared superconcepts, our approach goes into a similar direction as [6]. In contrast to earlier work, we employed well motivated measures of semantic similarity between the base concepts and their superconcepts. We conducted a series of experiments on the Reuters-21578 corpus using different sizes of training subsets and on the TREC question classification data. Our results indicate a consistent improvement in performance for superconcept semantic smoothing kernels in those cases where little training data is available or the feature representations are extremely sparse. Especially the lin scheme has proved to be a weighting scheme with stable improvements. As both [6] and [2] have pointed out, the success of semantic background knowledge in text-mining tasks critically depends on the employed word sense disambiguation strategy. Our experiments were deliberatly kept simple and did not use a word sense disambiguation step. While this effect is likely to have a negative impact on the results, the error introduced by this approach is systematic. In the light of these considerations, the results can also be seen as a pessimistic estimate of the potential effectiveness given a perfectly disambiguated input. As a different trail we will investigate the combination of our semantic kernels with other types of kernels that also exploit syntactic structure.

References

[1] R. Basili, M. Cammisa, and A. Moschitti. A Semantic Kernel to Classify Texts with Very Few Training Examples. In *Proc. Workshop 'Learning in Web Search', 22nd International Conference on Machine Learning (ICML 2005)*, 2005.

[2] S. Bloehdorn and A. Hotho. Text Classification by Boosting Weak Learners based on Terms and Concepts. In *Proc. 4th IEEE International Conference on Data Mining (ICDM 2004)*. IEEE Computer Society, 2004.

[3] A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[4] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent Semantic Kernels. *Journal of Intelligent Information Systems*, 18(2-3):127–152, 2002.

[5] T. Joachims. Making large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[6] D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *Proc. 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*. Springer, 2005.

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

[8] G. Siolas and F. d'Alché Buc. Support Vector Machines Based on a Semantic Kernel for Text Categorization. In *Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)*, 2000.

[9] D. Zhang and W. S. Lee. Question Classification Using Support Vector Machines. In *Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR 2003)*, 2003.