A Novel Higher-Order Semantic Kernel for Text Classification

Berna Altınel[#1], Murat Can Ganiz[#2], Banu Diri[#3]

[#]*Department of Computer Engineering, Marmara University, Istanbul, Turkey*

[1]berna.altinel@marmara.edu.tr

[#]*Department of Computer Engineering, Dogus University, Istanbul, Turkey*

[2]mcganiz@dogus.edu.tr

[#]*Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey*

[3]banu@ce.yildiz.edu.tr

***Abstract*— In conventional text categorization algorithms, documents are symbolized as "bag of words" (BOW) with the fact that documents are supposed to be independent from each other. While this approach simplifies the models, it ignores the semantic information between terms of each document. In this study, we develop a novel method to measure semantic similarity based on higher-order dependencies between documents. We propose a kernel for Support Vector Machines (SVM) algorithm using these dependencies which is called Higher-Order Semantic Kernel. With the aim of presenting comparative performance of Higher-Order Semantic Kernel we performed many experiments not only with our algorithm but also with existing traditional first-order kernels such as Polynomial Kernel, Radial Basis Function Kernel, and Linear Kernel. The experiments using Higher-Order Semantic Kernel on several well-known datasets show that classification performance improves significantly over the first-order methods.**

***Keywords*— Machine learning, support vector machine, text classification, higher order paths, semantic kernel.**

I. Introduction

It is one of the important ways to handle growing amounts of textual data on the social media and internet is to categorize documents into existing classes. Traditionally, text classification systems use the Bag of Words (BOW) method in order to represent the relationship between terms and documents in which each dimension is a term from the dictionary that consists of all documents in the corpus. As it is mentioned in [1] although it is not complicated and very popular, this representation has some drawbacks. First of all, it does not support multi-word phrases like "Support Vectors" since it locates these words into independent features; secondly it treats synonymous words as distinct components; and thirdly it maps polysemous words (i.e., words with multiple meanings) directly into one single component. Thus, in order to get higher classification accuracy it is vital to use conceptual patterns and semantic information. Kernel based methods, especially SVM is highly popular in text classification [3].

In this work, we present a new kernel for SVM called Higher-Order Semantic Kernel (HOSK) which is based on higher order paths for capturing hidden semantic information between documents for text classification. In the experiments, our proposed framework is compared with traditional kernels for SVM such as Linear Kernel, Radial Basis Function (RBF) Kernel and Polynomial Kernel. These traditional kernels can be considered as first-order methods.

II. Background and Related Work

A. Support Vector Machines

SVM was first presented by Vapnik, Guyon and Boser [12] in 1992**.** SVM maps the input data into a new space using kernel function by using support vectors and then performs a linear separation in this new space. Actually a kernel is a dot product in this new space (it is also called feature space or the Hilbert Space), which maps the input space's data into feature space. SVM tries to find a separating hyperplane between two classes of instances that has a *Maximum Margin* in the Hilbert space. SVM can be used for multi-class categorization, e.g., with the "one-against-the-rest" strategy [4]. Joachims showed that SVM produces good accuracies in the cases of like high-dimensionality and sparse data those are characteristics of text classification [13].

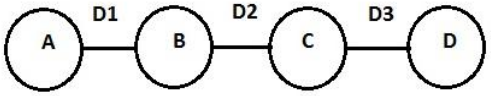
In our experiments we use WEKA [10] implementation of Sequential Minimal Optimization (SMO) [11] algorithm which solves the optimization problem that comes with the training part of SVM.

B. Higher Order Paths

There are several systems using higher-order co-occurrences in text classification. One of the most popular of them is the Latent Semantic Indexing (LSI) algorithm. Kontostathis et al. [5] proved mathematically that performance of LSI has a direct relationship with the higher-order paths. LSI's higher-order paths extract "latent semantics" [5], [6]. Based on these work, Ganiz et al. [2], [6] built a new Bayesian classification framework called Higher Order Naive Bayes (HONB) which presents that words in documents are strongly connected by such higher-order paths and they can be exploited in order to get better performance for classification.

Fig. 1 illustrates a higher-order-path which includes three documents with the names D1, D2 and D3, all of them contain

two of the following terms illustrated by the capitals A, B, C and D, respectively. As it is demonstrated in Fig. 1 term A is linked to term C through term B, so it makes a higher order path that links term A with term C. Because this path contains two edges it is called a third-order path [2]. Also there is a third-order path that links term A with term D through B and C. [2].



First order term co-occurrence {A, B}, {B, C}, {C, D}
Second order term co-occurrence {A, C}, {B, D}
Third order term co-occurrence {A, D}

Fig. 1 Higher Order term co-occurrences [2]

In our study we focused on the higher-order paths between documents rather than higher-order paths between terms.

C. Semantic Kernels for Text Classification

There are several studies which extracts semantic information between terms for kernel based classifications of documents. They usually employ external semantic sources. In particular WordNET [14], a hierarchical semantic database of English words, has been extensively used.

In one of these, Siolas et al. [7] created a semantic kernel based on WordNET, which could be seen as a semantic network, for obtaining term similarity information. In their work an estimation of two words' semantic relation is supplied by WordNET's hierarchical tree structure. Siolas et al. [7] have included this knowledge into the definition of Gaussian kernel. Their results show that the existence of semantic proximity metric increases the classification accuracy in SVM [7]. However, their approach treats multi-word concepts as single terms and does nothing to handle polysemy.

Semantic kernels with super concept declaration were studied in [8]. The aim of their work is to add the topological knowledge of their super concept expansion into the semantic kernel functions. However their experiments were kept introductory and did not use a word sense disambiguation strategy. [8]

Similarly, [7], [8] and [15] used WordNET as their ontology. On the other hand, Wang et al. [1] adds background knowledge extracted from Wikipedia [16] into a semantic kernel for enriching the representation of documents, which overcomes the shortages of the BOW approach. Their results demonstrate adding semantic knowledge into document representation by means of Wikipedia improves the categorization accuracy.

III. METHODOLOGY

In our system, matrix D is built from the whole corpus as a classical document by term frequency matrix. Let D be the data matrix having *r* rows (documents) and *c* columns (words) based on the whole corpus; m_{ij} shows occurrence frequency of the *jth* word in the *ith* document;

$\mathbf{m_i} = [m_{i1} .. m_{ic}]$ is the row vector representing the document *i* and $\mathbf{m^j} = [m_{1j} .. m_{rj}]$ the column vector corresponding to word *j*.

Since we deal with textual datasets with high dimensionality and sparsity, a proper normalization on this initial data matrix is beneficial. We tried many matrix normalization techniques from the literature which are detailed in [9] such as z-score normalization, min-max normalization, row-level normalization etc. We obtain best accuracy results with row-level normalization which is defined below:

$$\forall_i \in 1..r \quad D = \frac{m_i}{\max(m_i)} \tag{1}$$

where r is the number of documents in the corpus.

We then calculate first order paths matrix namely F between documents like in the Eq. 2:

$$F = D \times D^T \tag{2}$$

In other words, F matrix is calculated by multiplying document by term matrix by its transpose. Each value in the matrix of F shows the similarity between corresponding documents. For instance the similarity between m_i ($document_i$) and m_j ($document_j$) is calculated as follows:

$$F(i, j) = m_{i1} \times m_{j1} + m_{i2} \times m_{j2} + m_{i3} \times m_{j3} + ... + m_{ic} \times m_{jc} \tag{3}$$

where c is the number of terms in the corpus.

F is a (document by document) square matrix whose dimension is the same as the number of documents in the corpus. We observe that F has many zero values. Two documents have a non-zero similarity value in F only if these two documents share same words. In order to capture the latent semantic information between documents we calculated the second order paths matrix namely S between documents as in the Eq. 4:

$$S = F \times F \tag{4}$$

By multiplying F by itself S matrix is formed. Again S is a document by document square matrix whose dimension is the same as the number of documents in the corpus. This matrix shows the second order paths between documents.

Each value in the matrix of S shows the similarity between corresponding documents. For instance the similarity between m_i ($document_i$) and m_j ($document_j$) is calculated as in Eq. 5:

$$\begin{aligned} S(i, j) = {} & \\ & F(m_i, m_1) \times F(m_j, m_1) + F(m_i, m_2) \times F(m_j, m_2) + \\ & F(m_i, m_3) \times F(m_j, m_3) + F(m_i, m_4) \times F(m_j, m_4) + ... + \\ & F(m_i, m_r) \times F(m_j, m_r) \end{aligned} \tag{5}$$

So, both F and S have similarity information between documents. We observed that the values in S are extremely

larger than the ones in F. This could be explained with the reality that S has values not only based on only common terms between documents but also some indirectly latent semantics. So, we normalize the two matrix separately using Eq. 6 and Eq. 7:

$$\forall i, j \in 1...r \quad FN(m_i, m_j) = \frac{F(m_i, m_j)}{\max(F)} \tag{6}$$

$$\forall i, j \in 1...r \quad SN(m_i, m_j) = \frac{S(m_i, m_j)}{\max(S)} \tag{7}$$

where r is the number of documents in our corpus, F is first order paths matrix, FN is normalized first order paths matrix, S is second order paths matrix, and SN is normalized second order paths matrix, respectively. After this normalization we combine these two matrices with a weight value λ in order to see the effect of FN and SN matrices to the accuracy in our experiments. In order to optimize λ, the following values are taken into consideration: 0, 0.25, 0.5, 0.75, 0.8, 0.85, 0.9, 0.95, and 1. Combination of FN and SN matrices yield us Eq. 8 to build the final similarity matrix:

$$\mathrm{Sim}(d_i, d_j) = (\lambda \times SN_{ij}) + ((1 - \lambda) \times FN_{ij}) \tag{8}$$

Based on the results we tune the λ parameter to the value of 0.95.This is a satisfactory result for us from the aspect of the more contribution of higher-order paths means the more accurate results.

After that, we use this similarity matrix as a kernel (gram) matrix in SVM by plugging in the SMO WEKA's implementation. One of the most important parameter of SMO algorithm is misclassification-cost (C) parameter. After a set of optimization experiments we did not observe a significant difference and that's why we tuned it to its default value of 1.

IV. Experimental Setup and Experimental Results

A. Experimental Setup

In order to examine the performance of HOSK in SVM, we run it on several commonly used textual datasets. 1150Haber is our first dataset. It contains 1150 news-articles within five categories under the titles of magazine, politics, sport, economy and health collected from Turkish online newspapers [17]. Our second dataset is five-class version of the WebKB[19] dataset, namely WEBKB5, which contains web pages gathered from different universities' computer science departments. WebKB5 dataset has highly skewed class distribution. As the third dataset we used a variant of 20 Newsgroups dataset which is called 20News-18828[19].We used this dataset as in one of the basic subgroup form namely "SCIENCE".

We apply stemming and stopword filtering to the datasets. Terms occur less than three times in the documents are filtered. Furthermore, we used Information Gain in order to select most informative terms.

In order to see how our semantic kernel behaves under different conditions, we used the following percentage values for training set size; 1%, 5%, 10%, 30%, 50%, 70%, 80%, 90%. Remaining documents are used for testing. After running algorithms on 10 random splits for each of the above cases we calculate average of these 10 results as in [18]. We run our experiments using our experiment framework called Turkuaz, which closely uses WEKA library.

B. Experimental Results and Discussion

The main evaluation metric in our experiments is accuracy result and in the results tables they are written with their standard deviations. Also Student's t-Tests for statistical significance tests are provided. We use $\alpha = 0.05$ significance level which is a commonly used level. In addition for the accuracy we used the following performance gain equation;

$$Gain_{HOSK} = (P_{HOSK} - P_x) / P_x \tag{9}$$

where P_{HOSK} is the accuracy of SMO with HOSK and P_x stands for the accuracy result of the other kernels (linear, polynomial or RBF). The experimental results are demonstrated in Table I, Table II and Table III. These tables include training set percentage (TS), the accuracy results of Linear Kernel, Polynomial Kernel, and HOSK. Also the last columns demonstrate the (%) gain of HOSK over linear Kernel calculated as in Eq. 9.

According to our experiments HOSK results notable performance on 1150Haber dataset, which can be seen in Table I. HOSK outperforms our baseline kernel (Linear Kernel, which is one of the state-of-the-art kernels in text classification) by extensive boundaries in all training set percentages. The performance gain is specifically obvious at low training set levels. For instance at training levels 1%, 5%, and 10% HOSK statistically significantly outperforms Linear Kernel with the gains of 20.39% ,15.82%, 8.81% on Linear Kernel ,respectively.

TABLE I
ACCURACY OF DIFFERENT KERNELS ON 1150HABER DATASET WITH VARYING TRAINING SET SIZE

TS	Linear	Polynomial	RBF	HOSK	Gain
1	46.99±4.54	28.06±2.11	39.44±6.57	**56.57±12.22**	20.39
5	72.82±4.68	38.95±5.55	48.34±9.81	**84.34±2.33**	15.82
10	80.51±2.68	47.27±3.67	52.17±5.09	**87.60±1.26**	8.81
30	88.55±1.34	62.35±3.15	66.35±3.16	**90.78±0.58**	2.52
50	89.72±1.13	66.23±2.37	66.89±1.41	**90.90±0.82**	1.32
70	91.59±1.06	72.52±1.30	70.64±1.30	**92.41±0.54**	0.90
80	92.30±2.89	73.09±3.74	69.91±3.39	**92.43±3.15**	0.14
90	91.83±3.18	74.35±2.73	72.52±2.79	**92.17±2.21**	0.37

On WEBKB5 dataset, at training levels 1%, 5%, and 10% HOSK gives statistically significant results over Linear Kernel besides the results that HOSK outperforms than Linear Kernel in all of the training levels.

TABLE II
ACCURACY OF DIFFERENT KERNELS ON WEBKB5 DATASET WITH VARYING TRAINING SET SIZE

TS	Linear	Polynomial	RBF	HOSK	Gain
1	59.74±3.48	46.76±2.06	40.79±1.83	**75.1±2.82**	25.71
5	72.77±1.43	60.63±2.90	49.05±1.39	**84.20±0.87**	15.71
10	78.46±1.60	67.89±2.47	52.67±2.84	**86.79±0.62**	10.62
30	84.88±0.97	78.92±2.55	59.66±2.67	**88.45±0.46**	4.21
50	86.40±0.96	81.53±1.85	63.08±2.13	**89.11±0.48**	3.14
70	88.05±1.05	82.81±1.33	64.20±2.03	**89.61±0.65**	1.77
80	87.77±1.61	82.63±1.73	64.73±2.29	**89.68±1.01**	2.18
90	88.28±1.18	83.95±1.31	66.03±1.94	**89.15±1.56**	0.99

The performance improvement is most visible in small training set levels for instance in train split 1%, HOSK can achieve an accuracy of 72.59% where the Linear Kernel accuracy is only 52.16% for 20NewsSCIENCE dataset, which can be clearly seen from Table III.

TABLE III
ACCURACY OF DIFFERENT KERNELS ON 20NEWSSCIENCE DATASET WITH VARYING TRAINING SET SIZE

TS	Linear	Polynomial	RBF	HOSK	Gain
1	52.16±5.25	32.94±2.38	33.93±4.91	**72.59±5.84**	39.17
5	70.93±3.89	45.65±3.23	49.16±3.78	**85.69±1.80**	20.81
10	77.74±3.52	55.77±4.73	51.72±4.64	**87.87±1.34**	13.03
30	86.73±1.32	70.34±2.43	59.19±1.03	**93.11±0.77**	7.36
50	88.94±1.16	76.42±0.99	63.60±1.80	**94.18±0.48**	5.89
70	90.37±0.93	79.57±2.00	66.82±1.97	**95.07±0.86**	4.84
80	91.25±1.56	81.60±2.13	68.15±1.78	**95.4±0.87**	4.55
90	91.15±1.73	81.40±2.58	68.45±3.06	**96±1.80**	5.32

At small training data levels first order methods give zero as the similarity of two instances those do not contain common words. But by the use of higher order paths the similarity between those two instances can be larger than zero.

V. CONCLUSIONS

We have performed detailed experiments on several popular text-datasets and compared HOSK with traditional SVM kernels including state of the art Linear Kernel for text classification. Experiment results show that HOSK significantly outperforms the Linear Kernel, Polynomial Kernel, and RBF in all of our datasets under different training level conditions. Our results show the usefulness of HOSK as a semantic kernel for SVM in text classification.

As future work, we want to analyze the improved performance of HOSK. Especially, we would like to shed light into if and how our approach implicitly captures semantic information such as synonyms, and performs word sense disambiguation for polysemous terms when calculating similarity between documents. Also; we plan to get more insights about under which conditions HOSK performs well.

ACKNOWLEDGMENT

This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number 111E239. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of the TÜBİTAK.

REFERENCES

[1] P. Wang and C. Domeniconi, "Building Semantic Kernels for text classification using Wikipedia.", in Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 713-721, ACM Press, New York, 2008.

[2] M. Ganiz, N. Lytkin, W.M. Pottenger, "Leveraging Higher Order Dependencies between Features for Text Classification." in Proceedings of ECML/PKDD Conference, September, Bled, Slovenia , 2009.

[3] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[4] C.-W. Hsu, C.-J. Lin, "A Comparison of Methods for Multi-Class Support Vector Machines", IEEE Transactions on Neural Networks, pp. 415–425, 2002.

[5] A. Kontostathis and W.M. Pottenger,"A Framework for Understanding LSI Performance", in Information Processing & Management, pp. 56-73, 2006.

[6] M. Ganiz, C. George, W.M. Pottenger, "Higher Order Naive Bayes: A Novel Non-IID Approach to Text Classification", in IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 7, pp. 1022-1034, 2011.

[7] G. Siolas and F. D'Alche-Buc, "Support vectors machines based on a semantic kernel for text Categorization", in Proceedings of the International Joint Conference on Neural Networks, IEEE Press, Como , 2000.

[8] S. Bloehdorn, R. Basili, M. Cammisa and A. Moschitti, "Semantic kernels for text classification based on topological measures of feature similarity.", in ICDM '06: Proceedings of The Sixth International Conference on Data Mining, pp. 808–812, 2006.

[9] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Third Edition,2012

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, *The WEKA Data Mining Software*: An Update. SIGKDD Explorations, pp. 10-18, 2009.

[11] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", in Advances in Kernel Method: Support Vector Learning, MIT Press, pp. 185–208, 1998

[12] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifier", in Proc. 5th ACM Workshop, Comput. Learning Theory, pp. 144–152, Pittsburgh, 1992.

[13] T. Joachims, "Text Categorization with Many Relevant Features", in Proceedings of European Conference on Machine Learning, Springer Verlag, 1998.

[14] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Five Papers on Wordnet", Technical report, Stanford University ,1993.

[15] Q. LUO and E. Chen and H. Xiong,"A Semantic Term Weighting Scheme for Text Categorization", in Journal of Expert Systems with Applications, 2011.

[16] (2013) The Wikipedia website. [Online]. Available: http://www.wikipedia.org/

[17] M. F. Amasyali, and A. Beken, "Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi Ve Metin Siniflandirmada Kullanilmasi", in Proc IEEE Sinyal İşleme ve İletişim Uygulamalari Kurultayi (SIU), IEEE Press, 2009.

[18] M. Poyraz, Z.H. Kilimci, M.C. Ganiz," A Novel Semantic Smoothing Method Based on Higher Order Paths for Text Classification", in IEEE International Conference on Data Mining (ICDM), Brussels, Belgium , 2012.

[19] (2013) Two Text Learning Datasets Website [Online] Available:http://www.cs.cmu.edu/~textlearning