Engineering Applications of Artificial Intelligence 43 (2015) 54–66

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A corpus-based semantic kernel for text classification by using meaning values of terms

Berna Altınel [a,*], Murat Can Ganiz [b], Banu Diri [c]

[a] Department of Computer Engineering, Marmara University, İstanbul, Turkey
[b] Department of Computer Engineering, Doğuş University, İstanbul, Turkey
[c] Department of Computer Engineering, Yıldız Technical University, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 21 November 2014
Received in revised form
19 March 2015
Accepted 30 March 2015

Keywords:
Support vector machines
Text classification
Semantic kernel
Meaning
Higher-order relations

ABSTRACT

Text categorization plays a crucial role in both academic and commercial platforms due to the growing demand for automatic organization of documents. Kernel-based classification algorithms such as Support Vector Machines (SVM) have become highly popular in the task of text mining. This is mainly due to their relatively high classification accuracy on several application domains as well as their ability to handle high dimensional and sparse data which is the prohibitive characteristics of textual data representation. Recently, there is an increased interest in the exploitation of background knowledge such as ontologies and corpus-based statistical knowledge in text categorization. It has been shown that, by replacing the standard kernel functions such as linear kernel with customized kernel functions which take advantage of this background knowledge, it is possible to increase the performance of SVM in the text classification domain. Based on this, we propose a novel semantic smoothing kernel for SVM. The suggested approach is based on a meaning measure, which calculates the meaningfulness of the terms in the context of classes. The documents vectors are smoothed based on these meaning values of the terms in the context of classes. Since we efficiently make use of the class information in the smoothing process, it can be considered a supervised smoothing kernel. The meaning measure is based on the Helmholtz principle from Gestalt theory and has previously been applied to several text mining applications such as document summarization and feature extraction. However, to the best of our knowledge, ours is the first study to use meaning measure in a supervised setting to build a semantic kernel for SVM. We evaluated the proposed approach by conducting a large number of experiments on well-known textual datasets and present results with respect to different experimental conditions. We compare our results with traditional kernels used in SVM such as linear kernel as well as with several corpus-based semantic kernels. Our results show that classification performance of the proposed approach outperforms other kernels.

1. Introduction

Text categorization plays a significantly important role in recent years with the rapid growth of textual information on the web, especially on social networks, blogs and forums. This enormous data increases by the contribution of millions of people every day. Automatically processing these increasing amounts of textual data is an important problem. Text classification can be defined as automatically organizing documents into predetermined categories. Several text categorization algorithms depend on distance or similarity measures which compare pairs of text documents. For this reason similarity measures play a critical role in document classification. Apart from the other, structured data types, the textual data includes semantic information, i.e., the sense conveyed by the words of the documents. Therefore, classification algorithms should utilize semantic information in order to achieve better results.

In the domain of text classification, documents are typically represented by terms (words and/or similar tokens) and their frequencies. This representation approach is one of the most common one and it is called Bag of Words (BOW) feature representation. In this representation, each term constitutes a dimension in a vector space, independent of other terms in the same document (Salton and Yang, 1973). The BOW approach is very simple and commonly used; yet, it has a number of restrictions. Its main limitation is that it assumes independency between

* Corresponding author.
E-mail addresses: berna.altinel@marmara.edu.tr (B. Altınel), mcganiz@dogus.edu.tr (M. Can Ganiz), banu@ce.yildiz.edu.tr (B. Diri).

http://dx.doi.org/10.1016/j.engappai.2015.03.015
0952-1976/

terms, since the documents in BOW model are represented with their terms ignoring their position in the document or their semantic or syntactic connections between other words. Therefore it clearly turns a blind eye to the multi-word expressions by breaking them apart. Furthermore, it treats polysemous words (i.e., words with multiple meanings) as a single entity. For instance the term "organ" may have the sense of a body-part when it appears in a context related to biological structure, or the sense of a musical instrument when it appears in a context that refers to music. Additionally, it maps synonymous words into different components; as mentioned by Wang and Domeniconi (2008). In principle, as Steinbach et al. (2000) analyze and argue, each class has two types of vocabulary: one is "core" vocabulary which are closely related to the subject of that class, the other type is "general" vocabulary those may have similar distributions on different classes. So, two documents from different classes may share many general words and can be considered similar in the BOW representation.

In order to address these problems several methods have been proposed which use a measure of relatedness between term on Word Sense Disambiguation (WSD), Text Classification and Information Retrieval domains. Semantic relatedness computations fundamentally can be categorized into three such as knowledge-based systems, statistical approaches and hybrid methods which combine both ontology-based and statistical information (Nasir et al., 2013). Knowledge-based systems use a thesaurus or ontology to enhance the representation of terms by taking advantage of semantic relatedness among terms, for examples see (Bloehdorn et al., 2006), (Budanitsky and Hirst, 2006), (Lee et al., 1993), (Luo et al., 2011), (Nasir et al., 2013), (Scott and Matwin, 1998), (Siolas and d'Alché-Buc, 2000), and (Wang and Domeniconi, 2008). For instance in (Bloehdorn et al., 2006), (Siolas and d'Alché-Buc, 2000) the distance between words in WordNet (Miller et al., 1993) is used to capture semantic similarity between English words. The study in (Bloehdorn et al., 2006) uses super-concept declaration with different distance measures between words from WordNet such as Inverted Path Length (IPL), Wu-Palmer Measure, Resnik Measure and Lin Measure. A recent study of this kind can be found in (Zhang, 2013), which uses HowNet as a Chinese semantic knowledge-base. The second type of semantic relatedness computations between terms are corpus-based systems in which some statistical analysis based on the relations of terms in the set of training documents is performed in order to reveal latent similarities between them (Zhang et al., 2012). One of the famous corpus-based systems is Latent Semantics Analysis (LSA) (Deerwester et al., 1990) that partially solves the synonymy problem. Finally, approaches of the last category are called hybrid since they combine the information acquired both from the ontology and the statistical analysis of the corpus (Nasir et al., 2013), (Altınel et al., 2014a). There is a recent survey in (Zhang et al., 2012) about these studies.

In our previous studies, we proposed several corpus-based semantic kernels such as Higher-Order Semantic Kernel (HOSK) (Altınel et al., 2013), Iterative Higher-Order Semantic Kernel (IHOSK) (Altınel et al., 2014a) and Higher-Order Term Kernel (HOTK) (Altınel et al., 2014b) for SVM. In these studies, we showed significant improvements on classification performance over traditional kernels of SVM such as linear kernel, polynomial kernel and RBF kernel by taking advantage of higher-order relations between terms and documents. For instance, the HOSK is based on higher-order relations between the documents. The IHOSK is similar to the HOSK since they both propose a semantic kernel for SVM by using higher-order relations. However, IHOSK makes use of the higher-order paths between both the documents and the terms iteratively. Therefore, although, the performance of IHOSK is superior, its complexity is significantly higher than other higher-order kernels. A simplified model, the HOTK, uses higher-order paths between terms. In this sense, it is similar to the previously proposed term-based higher-order learning algorithms Higher-Order Naïve Bayes (HONB) (Ganiz et al., 2009) and Higher-Order Smoothing (HOS) (Poyraz et al., 2012, 2014).

In this article, we propose a novel approach for building a semantic kernel for SVM, which we name Class Meaning Kernel (CMK). The suggested approach smoothes the terms of a document in BOW representation (document vector represented by term frequencies) by class-based meaning values of terms. This in turn, increases the importance of significant or in other words meaningful terms for each class while reducing the importance of general terms which are not useful for discriminating the classes. This approach reduces the above mentioned disadvantages of BOW and improves the prediction abilities in comparison with standard linear kernels by increasing the importance of class specific concepts which can be synonymous or closely related in the context of a class. The main novelty of our approach is the use of this class specific information in the smoothing process of the semantic kernel. The meaning values of terms are calculated according to the Helmholtz principle from Gestalt theory (Balinsky et al., 2010, 2011a, 2011b, 2011c) in the context of classes.

We conducted several experiments on various document datasets with several different evaluation parameters especially in terms of the training set amount. Our experimental results show that CMK widely outperforms the performance of the other kernels such as linear kernel, polynomial kernel and RBF kernel. Please note that SVM with linear kernel is accepted as one the best performing algorithms for text classification and it virtually become de-facto standard in this domain. In linear kernel, the inner product between two document vectors is used as kernel function, which includes information about only the terms that these documents share. This approach can be considered as first-order method since its context or scope consists of a single document only. However, CMK can make use of meaning values of terms through classes. In this case semantic relation between two terms is composed of corresponding class-based meaning values of these terms for all classes. So if these two terms are important terms in the same class then the resulting semantic relatedness value will be higher. In contrast to the other semantic kernels that make use of WordNet or Wikipedia[1] in an unsupervised fashion, CMK directly incorporates class information to the semantic kernel. Therefore, it can be considered a supervised semantic kernel.

One of the important advantages of the proposed approach is its relatively low complexity. The CMK is a less complex and more flexible approach than the background knowledge-based approaches, since CMK does not require the processing of a large external knowledge base such as Wikipedia or WordNet. Furthermore, since CMK is constructed from corpus based statistics it is always up to date. Similarly, it does not have any coverage problem as the semantic relations between terms are specific to the domain of the corpus. This leads to another advantage of CMK: it can easily be combined with background knowledge-based systems that are using Wikipedia or WordNet. As a result, CMK outperforms other similar approaches in most of the cases both in terms of accuracy and execution time as can be seen from our experimental results.

The remainder of the paper is organized as follows: The background information with the related work including SVM, semantic kernels, and meaningfulness calculation summarized in Section 2. Section 3 presents and analyzes the proposed kernel for text classification algorithm. Experimental setup is described in Section 4, the corresponding experiment results including some discussion points are given in Section 5. Finally, we conclude the paper in Section 6 and provide a discussion on some probable future extension points of the current work.

[1] http://www.wikipedia.org/

2. Related work

2.1. Support vector machines for classification problem

Support Vector Machines (SVM) was first proposed by Boser et al. (1992). A more detailed analysis is given in (Vapnik, 1995). In general, SVM is a linear classifier that aims to finds the optimal separating hyperplane between two classes. The common representation of linearly separable space is

$$w^T\varphi(d)+b=0 \tag{1}$$

where w is a weight vector, b is a bias and d is the document vector to be classified. The problem of finding an optimal separating hyperplane can be solved by linearly constrained quadratic programming which is defined in the following equations:

$$\min \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i \tag{2}$$

with the constraints

$$y_i(w^T\varphi(d_i)+b)\geq 1-\xi_i$$

$$\xi_i\geq 0,\forall_i$$

where $\xi=(\xi_1,\xi_2...\xi_l)^T$ is the vector of slack variables and C is the regularization parameter, which is used to make a balance between training error and generalization, and has a critical role: if it is chosen as too large, there will be a high penalty for non-separable points, many support vectors will be stored, and the model will overfit; on the other hand if it is chosen too small, there will be underfitting (Alpaydın, 2004).

The problem in Eq. (2) can be solved using the Lagrange method (Alpaydın, 2004). After the solution the resultant decision function can be formulated as

$$f(x)=\mathrm{sgn}(\sum_{i=1}^{l}\alpha_i y_i k(d_i,d_j)+b) \tag{3}$$

where α_i is a Lagrange multiplier, k is a proper kernel function and samples $d_i$with $\alpha_i>0$ are called support vectors. An important property of a kernel function is that it has to satisfy Mercer's condition which means being semi-positive (Alpaydın, 2004). We can consider a kernel function as a kind of similarity function, which calculates the similarity values of data points, documents in our case, in the transformed space. Therefore, defining an appropriate kernel has the direct effect on finding a better representation of these data points as mentioned in Kontostathis and Pottenger (2006), Siolas and d'Alché-Buc (2000) and Wang and Domeniconi (2008). Popular kernel functions include linear kernel, polynomial kernel and RBF kernel:

- Linear kernel:

$$k(d_i,d_j)=d_i d_j \tag{4}$$

- Polynomial kernel:

$$k(d_i,d_j)=(d_i d_j+1)^q,\quad q=1,2...\text{etc.} \tag{5}$$

- RBF kernel:

$$k(d_i,d_j)=\exp(\gamma\|d_i-d_j\|^2) \tag{6}$$

For the problems of multiclass classification where there are more than two classes, a decomposition methodology is used to divide it into sub problems. There are basically two categories of multiclass methodology (Hsu and Lin, 2002): the all-in-one approach considers the data in one optimization formula (Wang et al., 2014), whereas the second approach is based on decomposing the original problem into several smaller binary problems, solving them separately and combining their solutions. There are two widely used basic strategies for this category: "one-against-the-rest" and "one-against-one" approaches (Dumais et al., 1998; Hsu and Lin, 2002).It is possible and common to use a kernel function in SVM which can map or transform the data into a higher dimensional feature space if it is impossible or difficult to find a separating hyperplane between classes in the original space; besides SVM can work very well on high dimensional and sparse data (Joachims, 1998). Because of these benefits of SVM, linear kernel is one of the best performing algorithms in text classification domain since textual data representation with BOW approach is indeed quite sparse and requires high dimensionality.

2.2. Semantic kernels for text classification

Linear kernel has been widely used in text classification domain since it is the simplest kernel function. As represented in Eq. (4) the calculated kernel values depend on the inner products of feature vectors of the documents. Mapping from input space to feature space is done with inner product. So a linear kernel captures similarity between documents as much as the words they share. This is a problem since it is not considering semantic relations between terms. This can be addressed by incorporating semantic information between words using semantic kernels as described in Altınel et al. (2013, 2014a, 2014b), Bloehdorn et al. (2006), Kandola et al. (2004), Luo et al. (2011), Nasir et al. (2011), Siolas and d'Alché-Buc (2000), Tsatsaronis et al. (2010), Wang and Domeniconi (2008) and Wang et al. (2014).

According to the definition mentioned in Alpaydın (2004), Bloehdorn et al. (2006), Boser et al. (1992), Luo et al. (2011) and Wang and Domeniconi (2008), any function in the following form Eq. (7), is a valid kernel function.

$$k(d_1,d_2)=\langle\varphi(d_1),\varphi(d_2)\rangle \tag{7}$$

In Eqs. (7), d_1 and d_2 are input space vectors and φ is a suitable mapping from input space into a feature space.

In Siolas and d'Alché-Buc (2000), the authors present a semantic kernel that is intuitively based on the semantic relations of English words in WordNet which is a popular and widely used network of semantic connections between words. These connections and hierarchies can be used to measure similarities between words. The authors use the distance between words in WordNet's hierarchical tree structure to calculate semantic relatedness between two words. They take advantage of this information to enrich the Gaussian kernel. Their results show that using the measured semantic similarities as smoothing metric increases the classification accuracy in SVM; but their approach ignores multi-word concepts as treating those single terms.

The study in Bloehdorn et al. (2006) uses super-concept declaration in semantic kernels. Their aim is to create a kernel algorithm which captures the knowledge of topology that belongs to their super-concept expansion. They utilize this mapping with the help of a semantic smoothing matrix Q that is composed of P and P^T which contains super-concept information about their corpus. Their suggested kernel function is given in Eq. (8). Their results demonstrate that they get notable improvement in performance, especially in situations where the feature representations are highly sparse or little training data exists (Bloehdorn et al., 2006).

$$k(d_1,d_2)=d_1PP^Td_2^T \tag{8}$$

In Bloehdorn and Moschitti (2007) a Semantic Syntactic Tree Kernel (SSTK) is built by incorporating syntactic dependencies such

as linguistic structures into a semantic knowledge that is gathered from WordNet. Similarly, in Kontostathis and Pottenger (2006) and Luo et al. (2011), WordNet is used as a semantic background information resource. However, they state that WordNet's coverage is not adequate and a wider background knowledge resource is needed. This is also one of the main reasons that other studies aim to use resources with wider coverage such as Wikipedia.

In one of these works, the authors combined the background knowledge gathered from Wikipedia into a semantic kernel for improving the representation of documents (Wang and Domeniconi, 2008). The similarity ratio between two documents in their kernel function formed as in Eq. (8), but in this case P is a semantic proximity matrix created from Wikipedia. The semantic proximity matrix is assembled from three measures. First of them is a content-based measure which depends on Wikipedia articles' BOW representation. Second measure is called the out-link-category-based measure that brings an information related to the out-link categories of two associative articles in Wikipedia. Third measure is a distance measure that is calculated as the length of the shortest path connecting the two categories of two articles belong to, in the acyclic graph schema of Wikipedia's category taxonomy. The authors claim that their method overcomes some of the shortages of the BOW approach. Their results demonstrate that adding semantic knowledge that is extracted from Wikipedia into document representation improves the categorization accuracy.

The study in Nasir et al. (2011) used semantic information from WordNet to build a semantic proximity matrix based on Omiotis (Tsatsaronis et al., 2010), which is a knowledge-based measure for computing the relatedness between terms. It actually depends on Sense Relatedness (SR) measure which discovers all the paths those connect a pair of senses in WordNet's graph hierarchy. Given a pair of senses s_1 and s_2, SR is defined as

$$SR(s_1, s_s) = \max_{P=(s_1,s_s)}\{SCM(P), SPE(P)\} \tag{9}$$

where P is a range over all the paths that connect s_1 to s_2, SCM and SPE are similarity measures depending on the depth of path's edges in WordNet. Nasir et al. (2013) also combined this measure into a Term Frequency-Inverse Document Frequency (TF-IDF) weighting approach. They demonstrate that their Omiotis-embedded methodology is better than standard BOW representation. Nasir et al. (2013) further extended their work by taking only top-k semantically related terms and using some evaluation metrics on larger text datasets.

The concept of Semantic Diffusion Kernel is presented by Kandola et al. (2004) and also studied by Wang et al. (2014). Such a kernel is obtained by an exponential transformation on a given kernel matrix as in

$$K(\lambda) = K_0 \exp(\lambda K_0) \tag{10}$$

where λ is the decay factor and K_0 is the gram or kernel matrix of the corpus in BOW representation. As mentioned in Wang et al. (2014) the kernel matrix K_0 is produced by

$$G = DD^T \tag{11}$$

where D is the feature representation of the corpus term by document. In Kandola et al. (2004) and Wang et al. (2014) it has been proved that $K(\lambda)$corresponds to a semantic matrix $\exp(\lambda\frac{G}{2})$as in the following:

$$S = \exp\left(\frac{\lambda}{2}G\right) = \frac{1}{2}\left(2I + \lambda G + \frac{\lambda^2 G^2}{2!} + \ldots + \frac{\lambda^\theta G^\theta}{0!} + \ldots\right) \tag{12}$$

where G is a generator which shows the initial semantic similarities between words and S is defined as the semantic matrix of the exponential of the generator. Wang et al. (2014) experimentally show that their diffusion matrix exploits higher-order co-occurrences to capture latent semantic relationships between terms in the WSD tasks from SensEval.

In our previous studies (Altınel et al., 2013, 2014a, 2014b) we built semantic kernels for SVM by taking advantages of higher-order paths. There are numerous systems with higher-order co-occurrences in text classification. One of the most widespread of them is the Latent Semantic Indexing (LSI) algorithm. The study in Kontostathis and Pottenger (2006) verified arithmetically that performance of LSI has a direct relationship with the higher-order paths. LSI's higher-order paths extract "latent semantics" (Ganiz et al., 2011; Kontostathis and Pottenger, 2006). Based on these work, the authors in Ganiz et al. (2009, 2011) 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 that they can be exploited in order to get better performance for classification. Both HONB (Ganiz et al., 2009) and HOS (Poyraz et al., 2012, 2014) are based on Naïve Bayes.

Benefits of using on higher-order paths between documents (Altınel et al., 2014a) and between terms (Altınel et al., 2014b; Ganiz et al., 2009; Poyraz et al., 2014) are demonstrated in Fig. 1. There are three documents, d_1, d_2, and d_3, which consist of a set of terms $\{t_1, t_2\}$, $\{t_2, t_3, t_4\}$, and $\{t_4, t_5\}$, respectively. Using a traditional similarity measure which is based on the common terms (e.g. dot product), the similarity value between documents d_1 and d_3 will be zero since they do share any terms. But this measure is misleading since these two documents have some connections in the context of the dataset over d_2 (Altınel et al., 2014b) as it can be perceived in Fig. 1. This supports the idea that using higher-order paths between documents, it is possible to obtain a non-zero similarity value between d_1 and d_3 which is not possible in the BOW representation. This value turns out to be larger if there are many interconnecting documents like d_2 between d_1 and d_3. This is caused by the fact that the two documents are written on the same topic using different but semantically closer sets of terms.

In Fig. 1, there is also a higher-order path between t_1 and t_3. This is an illustration of a novel second-order relation since these two terms do not co-occur in any of these documents and can remain undetected in traditional BOW models. However, we know that t_1 co-occurs with t_2 in document d_1, and t_2 co-occurs with t_3 in document d_2. The same principle that is mentioned in the case of documents above applies in here. The similarity between t_1 and t_3 becomes more eminent if there are many interconnecting terms such as t_2 or t_4 and interconnecting documents like d_2. The regularity of these second order paths may reveal latent semantic relationships such as synonymy (Poyraz et al., 2014).

In our previous study, we proposed a semantic kernel called Higher-Order Semantic Kernel (HOSK) which makes use of higher-order paths between documents (Altınel et al., 2013). In HOSK, a simple dot product between the features of the documents gives a first-order matrix **F**, where its second power, the matrix **S** reveals second-order relations between documents. The **S** is used as kernel smoothing matrix in HOSK's transformation from input space into feature space. The results show that HOSK gains an improvement on accuracy over not only linear kernel but also polynomial kernel and RBF. Based on this, a more advanced method called Iterative Higher-Order Semantic Kernel (IHOSK) is proposed in Altınel et al. (2014a). The IHOSK makes use of higher-order paths between documents and terms in an iterative algorithm. This study is inspired from the similarity measure developed in Bisson and Hussain (2008). Two similarity matrices, similarity between terms (**SC**) and similarity between documents (**SR**) are produced iteratively (Altınel et al., 2014a; Bisson and Hussain, 2008) using the following formulas:

$$SR_t = DSC_{t-1}D^T \bullet NR \text{ with } NR_{i,j} = \frac{1}{|d_i||d_j|} \tag{13}$$

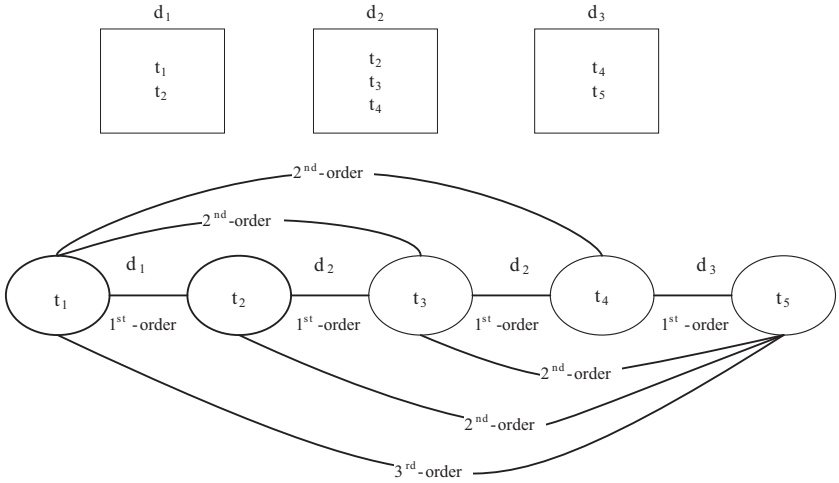

Fig. 1. Graphical demonstration of first-order, second-order and third-order paths between terms through documents (Altınel et al., 2014b). 1st-order term co-occurrence $\{t_1, t_2\}$, $\{t_2, t_3\}$, $\{t_3, t_4\}$, $\{t_2, t_4\}$, $\{t_4, t_5\}$; 2nd-order term co-occurrence $\{t_1, t_3\}$, $\{t_1, t_4\}$, $\{t_2, t_5\}$, $\{t_3, t_5\}$; 3rd-order term co-occurrence $\{t_1, t_5\}$.

$$SC_t = D^T SR_{t-1} D \bullet NC \quad \text{with } NC_{i,j} = \frac{1}{|d_i||d_j|} \tag{14}$$

where D is the document by term matrix, D^T is the transpose of D matrix, SR is the row (document) similarity matrix, SC is the column (word) similarity matrix, and NR and NC are row and column normalization matrices, respectively. Bisson and Hussain (2008) state that they repeat SR_t and SC_t calculations up to a limited number of iterations such as four. Based on our optimization experiments we tuned this number to two (Altınel et al., 2014a). After calculating SC_t, it is used in the kernel function for transforming instances from original space to feature space like in the following:

$$k_{IHOSK}(d_1, d_2) = d_1 SC_t SC_t^T d_2^T \tag{15}$$

where k_{IHOSK} is the kernel function value of documents d_1 and d_2, respectively.

According to the experiment results, the classification performance improves over the well-known traditional kernels used in the SVM such as the linear kernel, the polynomial kernel and RBF kernel.

In our most recent effort we consider less complex higher-order paths: the Higher-Order Term Kernel (HOTK) is based on the outcomes of higher-order paths between the terms only. The semantic kernel transformation in HOTK is done using the following equation:

$$k_{HOTK}(d_1, d_2) = d_1 SS^T d_2^T \tag{16}$$

where S contains higher-order co-occurrence relationships between terms in the training set only. HOTK is much simpler than IHOSK (Altınel et al., 2014a) from the points of implementation, combining with normalization or path-filtering techniques and also requires less computation time and less usage of memory resources.

2.3. Term weighting methods

TF-IDF is one of the common term weighting approaches and was proposed in Jones (1972). Its formula is given in (18), where tf_w represents the frequency of the term w in the document and *IDF* is the inverse of the document frequency of the term in the dataset. *IDF*'s formula is also given in Eq. (17) where $|D|$ denotes the number of documents and df_w represents the number of documents which contains term w. *TF* indicates the occurrence of word w in document d_i. TF-IDF has proved extraordinarily robust and difficult to beat, even by much more carefully worked out models and theories (Robertson, 2004).

$$\text{IDF}(w) = \frac{|D|}{df_w} \tag{17}$$

$$\text{TF} - \text{IDF}(w, d_i) = tf_w \times \log(\text{IDF}(w)) \tag{18}$$

A similar but supervised version of TF-IDF is called TF-ICF (Term Frequency – Inverse Class Frequency), whose formula given in Eq. (20) as in Ko and Seo (2000) and Lertnattee and Theeramunkong (2004). In Eq. (19), $|C|$ indicates number of classes and cf_w shows the number of classes which contain term w. It is simply calculated by dividing the total number of classes to the number of classes that this term w occurs in classes.

$$\text{ICF}(w) = \frac{|C|}{cf_w} \tag{19}$$

$$\text{TF} - \text{ICF}(w, c_j) = \sum_{d \in c_j} tf_w \times \log(\text{ICF}(w)) \tag{20}$$

2.4. Helmholtz principle from Gestalt theory and its applications to text mining

According to Helmholtz principle from Gestalt theory in image processing; "observed geometric structure is perceptually meaningful if it has a very low probability to appear in noise" (Balinsky et al., 2011a). This means that events that have a large deviation from randomness or noise can be noticed easily by humans. This can be illustrated in Fig. 2. In the left hand side of Fig. 2, there is a group of five aligned dots but it is not easy to notice it due to the high noise. Because of the high noise, i.e. large number of randomly placed dots, the alignment probability of five dots increases. On the other hand, if we remove the number of randomly placed dots considerably, we can immediately perceive

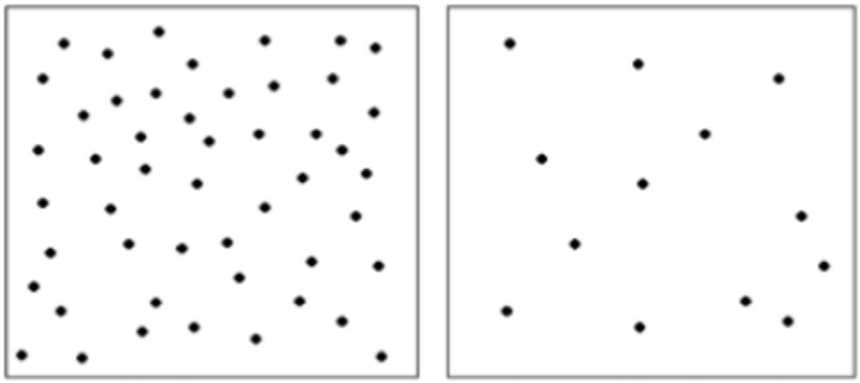

Fig. 2. The Helmholtz principle in human perception (adopted from Balinsky et al. (2011a)).

the alignment pattern in the right hand side image since it is very unlikely to happen by chance. This phenomenon means that unusual and rapid changes will not happen by chance and they can be immediately perceived.

As an example, assume you have unbiased coin and it is tossed 100 times. Any 100-sequence of heads and tails can be generated with probability of (½)100 and Fig. 3 is generated where 1 represents heads and 0 represents tails (Balinsky et al., 2010).

First sequence, s_1 is expectable for unbiased coin but second output, s_2 is highly unexpected. This can be explained by using methods from statistical physics where we observe macro parameters but we don't know the particular configuration. For instance expectation calculations can be used for this purpose (Balinsky et al., 2010).

A third example is known as birthday paradox in literature. There are 30 students in a class and we would like to calculate the probability of two students having the same birthday and how likely or interesting is this. Firstly, we assume that birthdays are independent and uniformly distributed over the 365 days of a year. Probability P_1 of all students having different birthday in the class is calculated in Eq. (21) (Desolneux et al., 2008).

$$P_1 = \frac{365x364x\ldots x336}{365^{30}} \approx 0.294 \tag{21}$$

The probability P_2 of at least two students born on same day is calculated in Eq. (22).This means that approximately 70% of the students can have the same birthday with another student in the class of 30 students.

$$P_2 = 1 - 0.294 = 0.706 \tag{22}$$

When probability calculations are not computable, we can compute expectations. The expectation of number of 2-tuples of students in a class of 30 is calculated as in Eq. (23). This means that on the average, 1.192 pairs of students have the same birthday in the class of 30 students and therefore it is not unexpected. However the expectation values for 3 and 4 students having the same birthday, $E(C3) \approx 0.03047$ and $E(C4) \approx 0.00056$, which are much smaller than one, indicates that these events will be unexpected (Desolneux et al., 2008).

$$E(C_2) = \frac{1}{365^{2-1}}\binom{30}{2} = \frac{1}{365}\frac{30!}{(30-2)!2!} = \frac{30x29}{2x365} \approx 1.192 \tag{23}$$

In summary, the above principles indicate that meaningful features and interesting events appears in large deviations from randomness. Meaningfulness calculations basically correspond to calculations of expectations and they stem from the methods in statistical physics (Balinsky et al., 2011a).

In the context of text mining, the textual data consist of natural structures in the form of sentences, paragraphs, documents, and topics. In (Balinsky et al., 2011a), the authors attempt to define

$$s_1 = 10101\ 11010\ 01001\ \ldots\ 00111\ 01000\ 10010$$

$$s_2 = \underbrace{111111111\ \ldots\ 111111}_{50\ times}\ \underbrace{000000000\ \ldots\ 000000}_{50\ times}$$

Fig. 3. The Helmholtz principle in human perception (adopted from Balinsky et al. (2010)).

meaningfulness of these natural structures using the human perceptual model of Helmholtz principle from Gestalt Theory. Modelling the meaningfulness of these structures is established by assigning a meaning score to each word or term. Their new approach to meaningful keyword extraction is based on two principles. The first one states that these keywords which are representative of topics in a data stream or corpus of documents should be defined not only in the document context but also the context of other documents. This is similar to the TF-IDF approach. The second one states that topics are signaled by "unusual activity", a new topic can be detected by a sharp rise in the frequencies of certain terms or words. They state that sharp increase in frequencies can be used in rapid change detection. In order to detect the change of a topic or occurrence of new topics in a stream of documents, we can look for bursts on the frequencies of words. A burst can be defined as a period of increased and unusual activities or rapid changes in an event. A formal approach to model "bursts" in document streams is presented in (Kleinberg, 2002). The main intuition in this work is that the appearance of a new topic in a document stream is signaled by a "burst of activity" with certain features rising sharply in frequency as the new topic appears.

Based on the theories given above, new methods are developed for several related application areas including unusual behavior detection and information extraction from small documents (Dadachev et al., 2012), for text summarization (Balinsky et al., 2011b), defining relations between sentences using social network analysis and properties of small world phenomenon (Balinsky et al., 2011c) and rapid change detection in data streams and documents (Balinsky et al., 2010) and also for keyword extraction and rapid change detection (Balinsky et al., 2011a). These approaches make use of the fact that meaningful features and interesting events come into view if their deviations from randomness are very large.

The motivating question in these studies is "if the word w appears m times in some documents is this an expected or unexpected event?" (Balinsky et al., 2011a). Given that S_w is the set of all words in N documents and a particular word w appears K times in these documents. Then random variable C_m counts m-tuple of the elements of S_w appears in the same document. Following this the expected value of C_m is calculated under the assumption that the words are independently distributed among the documents. C_m is calculated using random variable $X_{i1,i2\ldots im}$ which indicates if words $w_{i1},\ldots,w_{im}$ co-occurs in the same

document or not. Based on this the expected value $E(C_m)$ can be calculated as in Eq. (25) by summing the expected values of all these random variables for all the words in the corpus.

$$C_m = \sum_{1 \le i1 < \ldots < im \le K} X_{i1,\ldots,im} \tag{24}$$

$$E(C_m) = \sum_{1 \le i1 < \ldots < im \le K} E(X_{i1,\ldots,im}) \tag{25}$$

The random variable $X_{i1,i2\ldots im}$ can only take values one and zero. As a result the expectation of this random variable which shows if these m words co-occurs in the same document can be calculated in Eq. (26), where N is the total number of documents. "If in some documents the word w appears m times and $E(Cm) < 1$ then it is an unexpected event" (Balinsky et al., 2011a).

$$E(X_{i1,\ldots,im}) = \frac{1}{N^{m-1}} \tag{26}$$

As a result $E(Cm)$ can simply be expressed as in Eq. (27) and this expectation actually corresponds to Number Of False Alarms (NFA) of m-tuple of word w which is given in Eq. (28). This corresponds to the number of times m-tuple of the word w occurs by chance (Balinsky et al., 2011a). Based on this, in order to calculate the meaning of a word w which occurs m times in a context (document, paragraph, sentence), we can look its NFA value. If the NFA (expected number) is less than one, then the occurrence of m times can be considered as a meaningful event because it is not expected by our calculations but it is already happened. Therefore, word w can be considered as a meaningful or important word in the given context.

$$E(C_m) = \binom{K}{m} \frac{1}{N^{m-1}} \tag{27}$$

Based on the NFA, the meaning score of words are calculated using Eq. (28) and Eq. (29) in Balinsky et al. (2011c):

$$\mathrm{NFA}(w, P, D) = \binom{K}{m} \frac{1}{N^{m-1}} \tag{28}$$

$$\mathrm{Meaning}(w, P, D) = -\frac{1}{m} \log\ \mathrm{NFA}(w, P, D) \tag{29}$$

where w represents a word, P represents a part of the document such as a sentence or a paragraph, and D represents the whole document. Additionally, m indicates the appearance number of word w in P and K shows the appearance number of word w in D. $N = L/B$ in which L is the length of D and B is the length of P in words (Balinsky et al., 2011c). To define *Meaning* function, the logarithmic value of NFA is used based on the observation that NFA values can be exponentially large or small (Balinsky et al., 2011a).

As mentioned above, the meaning calculations are performed in a supervised setting. In other words, we use a class of documents as our basic unit or context in order to calculate meaning scores for words. In this approach meaning calculations basically show how high a particular words' frequency is expected to be in a class of documents compare to the other classes of documents. If it is unexpected then meaning calculations result in a high meaning score. In this aspect it is similar to the Multinomial Naïve Bayes in which the all the documents in a class are merged into a single document and then the probabilities are estimated from this one large class document. It also bears similarities to TF-ICF approach in which the term frequencies are normalized using the class frequencies.

In supervised meaning calculations, which are given in Eqs. (34) and (35), parameter c_j represents documents which belong to class j and S represents the complete training set. Assume that a feature w appears k times in the dataset S, and m times in the documents of class c_j. The length of dataset (i.e. training set) S and class c_j measured by the total term frequencies is L and B respectively. N is the ratio of the length of the dataset and the class, which is calculated in Eq. (32). The number of false alarms (NFA) is defined in Eq. (33).

$$L = \sum_{d \in S} \sum_{w \in d} tf_w \tag{30}$$

$$B = \sum_{d \in c_j} \sum_{w \in d} tf_w \tag{31}$$

$$N = \frac{L}{B} \tag{32}$$

$$NFA(w, c_j, S) = \binom{k}{m} \frac{1}{N^{m-1}} \tag{33}$$

Based on NFA, the meaning score of the word w in a class c_j is defined as:

$$meaning(w, c_j) = -\frac{1}{m} \log\ NFA(w, c_j, S) \tag{34}$$

This formula can be re-written as:

$$meaning(w, c_j) = -\frac{1}{m} \log \binom{k}{m} - [(m-1) \log\ N] \tag{35}$$

The larger the meaning score of a word w in a class c_j, the more meaningful, significant or informative that word is for that class.

3. Class Meanings Kernel (CMK)

In our study, we use the general form of kernel function which is given in Eq. (7). The simplest form of kernel function, namely linear kernel is formulated in Eq. (4). But as it is criticized in the previous section the linear kernel is a simple dot product between the features of text documents. It produces a similarity value of two documents only proportional to the number of shared terms. Combined with the highly sparse representation of the textual data, this may yield a significant problem especially when two documents are written about the same topic using two different sets of terms which are actually semantically very close as it is mentioned in the Section 2.2. We attempt to illustrate this using an extreme example in Fig. 1, where documents d_1 and d_3 do not share any common words. So, their similarity calculation will be zero if it is based on only the number of common words. But as it can be noticed from Fig. 1, without any controversy d_1 and d_3 have some similarity value through d_2, which is greater than zero. Also, in cases where training data is scarce there will be serious problems to detect reliable patterns between documents. This means that using only simple dot product to measure similarity between documents will not always give sufficiently accurate similarity values between documents. Additionally; as mentioned before, for a better classification performance it is inevitably required to discount general words and emphasize more importance on core words (which are closely related to the subject of that class) as is analyzed in (Steinbach et al., 2000). In order to overcome these mentioned drawbacks, semantic smoothing kernels encode semantic dependencies between terms (Basili et al., 2005; Bloehdorn et al., 2006; Mavroeidis et al., 2005; Siolas and d'Alché-Buc, 2000). We also incorporated additional information of terms other than their simple frequencies as in our previous studies (Altınel et al., 2013, 2014a, 2014b) in which we take advantage of higher-order paths between words and/or documents. In those studies we showed that the performance difference between first-order and higher-order representation of features. In this paper we investigate the use of a new type of semantic smoothing kernel for text.

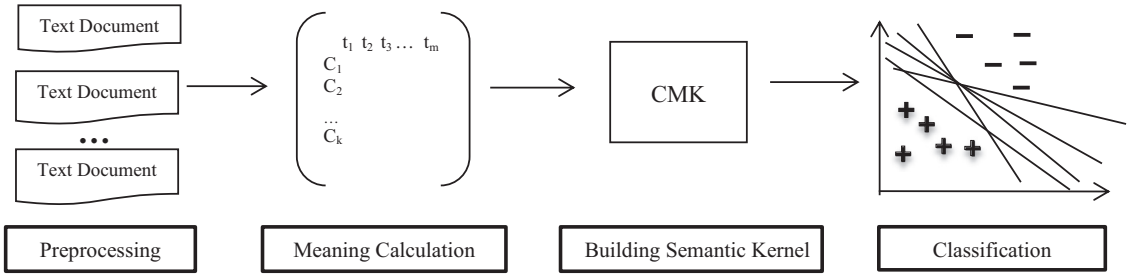

Fig. 4. The architecture of CMK system.

Fig. 4 demonstrates the architecture of the suggested semantic kernel. This system mainly consists of four independent modules: preprocessing, meaning calculation, building semantic kernel, and classification. Preprocessing is the step that involves the conversion of input documents into formatted information. This step's details (stemming, stopword filtering) will be described in Section 4. In meaning calculation step, the meaning values of the terms according to the classes are calculated based on Eq. (34). Then we construct our proposed kernel, namely CMK, in the step for building semantic kernel. Finally, in the classification step SVM classifier builds a model in the training phase and this model is then applied to the test examples in the test phase.

Clearly, the main feature of this system is that it takes advantages of the meaning calculation in kernel building process, in order to reveal semantic similarities between terms and documents by smoothing the similarity and the representation of the text documents. Meaning calculation is based on Helmholtz principle from Gestalt theory. As mentioned in Section 2.4, this meaning calculations have been applied to many domains in previous works (for example information extraction (Dadachev et al., 2012), text summarization (Balinsky et al., 2011b), rapid change detection in data streams (Balinsky et al., 2010), and keyword extraction). In these studies a text document is modelled by a set of meaningful words together with their meaning scores. A word is considered meaningful or important if the term frequency of a word in a document is unexpected if we consider the term frequencies of this word in all the documents in our corpus. The method can be applied on a single document or on a collection of documents to find meaningful words inside each part or context (paragraphs, pages, sections or sentences) of a document or a document inside of a collection of documents (Balinsky et al., 2011c). Although meaning calculation has been used in several domains, to the best of our knowledge, our work is the first to apply this technique to kernel function.

In our methodology $\boldsymbol{D_{train}}$ is the data matrix of training set having r rows (documents) and t columns (terms). In this matrix d_{ij} stands for the occurrence frequency of the jth word in the ith document; $d_i=[d_{i1},\ldots,d_{it}]$ is the document vector showing the document i and $d_j=[d_{1j},\ldots,d_{rj}]$ is the term vector belonging to word j, respectively. To enrich D_{train}, with semantic information, we build the class-based term meaning matrix M using meaning calculations given in Eq. (29). The M matrix shows the meaningfulness of the terms in each class. Based on M we calculate S matrix in order to reveal class based semantic relations between terms. Specifically, the i, j element of S quantifies the semantic relatedness between terms t_i and t_j.

$$S = MM^T \tag{36}$$

In our system S is a semantic smoothing matrix to transform documents from input space to feature space. Thus, S is a symmetric term-by-term matrix. Mathematically, the kernel value between two documents is given as

$$k_{CMK}(d_1, d_2) = d_1 SS^T d_2^T \tag{37}$$

where $k_{CMK}(d_1, d_2)$ is the similarity value between documents d_1 and d_2, S is the semantic smoothing matrix. In other words, here S is a semantic proximity matrix which derives from the meaning calculations of terms and classes.

If a word occurs only once in a class then its meaning value for that class is zero according to Eq. (29). If a word does not occur at all in a class, it gets minus infinity based on Eq. (29) as a meaning value for that class. In order to make calculations more practical we assign the next smallest value to that word according to the range of meaning values we get for all the words in our corpus. After all calculations we get M as a term-by-class matrix which includes the meaning values of terms in all classes of the corpus. We observe that these meaning values are high for those words that allow us to distinguish between classes. Indeed terms semantically close to the theme discussed in the documents of that class gain the highest meaning values in the range. In other words semantically related terms of that class, i.e. "core" words like it is mentioned in (Steinbach et al., 2000), gain importance while semantically isolated terms, i.e. "general" words lose their importance. So terms are ranked based on their importance. For instance, if the word "data" is highly present while the words "information" and "knowledge" are less, the application of semantic smoothing will increase the values of the last two terms because "data", "information" and "knowledge" are strongly related concepts. The new encoding of the documents is richer than the standard TF-IDF encoding since; additional statistical information that is directly calculated from our training corpus is embedded into the kernel. In other words transformations in Eq. (37) smooth the basic term vector representation using semantic ranking while passing from the original input space to a feature space through kernel transformation functions $\varphi(d_1)$ and $\varphi(d_1)$ for the documents d_1 and d_2 respectively:

$$\varphi(d_1) = d_1 S \text{ and } \varphi(d_2) = S^T d_2^T \tag{38}$$

As mentioned in (Wittek and Tan, 2009), the presence of S in Eq. (38) changes the orthogonality of the vector space model, as this mapping introduces term dependence. Documents can be seen as similar even if they do not share any terms by eliminating orthogonality.

Also as it is mentioned in (Balinsky et al., 2010), meaning calculation automatically filters stop words by assigning them very small amounts of meaning values. Let us consider the following two cases, which are represented in Table 1. According to Table 1, it is understood that t_1 and t_2 occurred in one or more documents of c_1, not in remaining classes; c_2, c_3 and c_4, respectively. In other words t_1 and t_2 are critical words of the topic discussed in c_1; getting high meaning values according to Eq. (29); since the frequency of a term in a class, m, is inversely proportional to the NFA. According to Eq. (29), in such a case the number of times that word occurred in the whole corpus (k) is larger when the times of that word's occurrence in a class (m) is smaller NFA calculation directly gives a larger negative value which will yield a larger positive value. In other words, according to the spirit of meaning value calculation, the more a word occurred in only a

Table 1
Term frequencies in different classes.

	c_1	c_2	c_3	c_4
t_1	1	0	0	0
t_2	1	0	0	0
t_3	1	1	1	1
t_4	1	1	1	1

specific class the higher meaning value it gets, and conversely the more a word occurred in all classes the less meaning value it gets. This statement can also be represented with Table 1, since t_1 and t_2 occurred in only c_1 while t_3 and t_4 occurred in every classes of the corpus. It is highly possible that these two words, t_3 and t_4, are in the type of "general" words since they are seen in every class of the corpus.

4. Experiment setup

We integrated our kernel function into the implementation of the SVM algorithm in WEKA (Hall et al., 2009). In other words, we built a kernel function that can be directly used with Platt's Sequential Minimal Optimization (SMO) classifier (Platt, 1998).

In order to see the performance of CMK on text classification, we performed a series of experiments on several textual datasets which are shown in Table 2. Our first dataset IMDB[2] is a collection of movie reviews. It contains 2000 reviews about several movies in IMDB. There are two types of labels; *positive* and *negative*. The labels are balanced in both training and test sets that we used in our experiments. Other datasets are variants of popular 20 Newsgroup[3] dataset. This data set is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups and commonly used in machine learning applications, especially for text classification and text clustering. We used four basic subgroups "POLITICS", "COMP", "SCIENCE", and "RELIGION from the 20 Newsgroup dataset. The documents are evenly distributed to the classes. The sixth dataset we use is the mini-newsgroups[4] dataset which has 20 classes and also has a balanced class distribution. This is a subset of the 20 Newsgroup[2] dataset, too. Properties of these datasets are given in Table 2.

We apply stemming and stopword filtering to these datasets. Additionally, we filter rare terms which occur in less than three documents. We also apply attribute selection and select the most informative 2,000 terms using Information Gain as described in Altınel et al. (2013, 2014a, 2014b), Ganiz et al., 2009, (2011) and Poyraz et al. (2012, 2014). This preprocessing increase the performance of the classifier models by reducing the noise. We perform this preprocessing equally in all experiments we report in the following.

In order to observe the behavior of our semantic kernel under different training set size conditions, we use the following percentage values for training set size: 5%, 10%, 30%, 50%, 70%, 80% and 90%. Remaining documents are used for testing. This is essential since we expect that the advantage of using semantic kernels should be more observable when there is inadequate labeled data.

One of the main parameters of SMO (Kamber and Frank, 2005) algorithm is the misclassification cost (C) parameter. We conducted a series of optimization experiments on all of our datasets with the values of $\{10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}\}$. For all the training set percentages we selected the best performing one. The optimized C values for each dataset at different training levels are given in Table 3. This is interesting because the values vary a lot among datasets and training set percentages (TS).

Table 2
Properties of datasets before attribute selection.

Dataset	#classes	#instances	#features
IMDB	2	2000	16,679
20News-POLITICS	3	1500	2478
20NewsGroup-SCIENCE	4	2000	2225
20News-RELIGION	4	1500	2125
20News-COMP	5	2500	2478
Mini-NewsGroups	20	2000	12,112

Table 3
Optimized C values for our datasets.

TS%	IMDB	SCIENCE	POLITICS	RELIGION	COMP	Mini-newsgroups
5	0.01	0.01	0.01	0.01	0.10	10.0
10	0.01	0.01	0.01	0.01	0.10	100
30	0.10	0.01	0.01	0.01	1.00	10.0
50	0.01	0.10	0.01	0.10	100	1.00
70	10.0	0.10	0.10	0.01	0.10	1.00
80	0.01	0.01	0.01	0.10	0.10	1.00
90	0.01	0.01	0.01	0.01	100	1.00

After running algorithms on 10 random splits for each of the training set ratios with their optimized C values, we report average of these 10 results as in (Altınel et al., 2014a, 2014b). This is a more comprehensive way of well-known n-fold cross validation which splits the data into n sets and train on n-1 of them while the remaining used as test set. Since the training set size in this approach is fixed (for instance it is 90% for 10-fold cross validation) we cannot analyze the performance of the algorithm under scarcely labeled data conditions.

The main evaluation metric in our experiments is the accuracy and in the results tables we also provide standard deviations.

In order to highlight the performance differences between baseline algorithms and our approach we report performance gain calculated using the following equation:

$$\text{Gain}_{CMK} = \frac{(P_{CMK} - P_x)}{P_x} \tag{39}$$

where P_{CMK} is the accuracy of SMO with *CMK* and P_x stands for the accuracy result of the other kernel. The experimental results are demonstrated in Table 4–9. These tables include training set percentage (TS), the accuracy results of linear kernel, polynomial kernel, RBF kernel, IHOSK, HOTK and CMK. Also the "Gain" columns in the corresponding results tables demonstrate the (%) gain of CMK over linear kernel calculated as in Eq. (39). Additionally, Students t-Tests for statistical significance are provided. We use $\alpha = 0.05$ significance level which is a commonly used level. In the training sets, where CMK significantly differs over linear kernel based on Students t-Tests, we indicate this with "*". Furthermore we also provide the term coverage ratio by;

$$\text{Term Coverage} = \frac{n}{N} \times 100 \tag{40}$$

where n is the number of different terms seen in the documents of training set percentages and N is the total number of different terms in our corpus; respectively. We observe a reasonable relevance between the accuracy differences and term coverage ratios while passing from one training set percentage to another, which will be discussed in the following section in a detailed way.

[2] http://www.imdb.com/interfaces
[3] http://www.cs.cmu.edu/~textlearning
[4] http://archive.ics.uci.edu/ml/

Table 4
Accuracy of different HO kernels on SCIENCE dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	71.44 ± 4.3	45.65 ± 3.23	49.16 ± 3.78	**84.15 ± 2.87**	76.63 ± 2.67	64.51 ± 4.86	−9.70	63.99
10	77.97 ± 3.73	55.77 ± 4.73	51.72 ± 4.64	**90.37 ± 0.81**	82.47 ± 2.02	82.19 ± 3.58	5.41*	82.28
30	86.73 ± 1.32	70.34 ± 2.43	59.19 ± 1.03	94.31 ± 1.09	89.24 ± 0.74	**95.07 ± 0.87**	9.62*	98.01
50	88.94 ± 1.16	76.42 ± 0.99	63.60 ± 1.80	94.97 ± 0.90	90.84 ± 1.12	**96.71 ± 0.61**	8.74*	99.90
70	90.58 ± 0.93	79.57 ± 2.00	66.82 ± 1.97	95.35 ± 0.88	92.06 ± 1.28	**97.12 ± 0.59**	7.22*	99.99
80	91.33 ± 1.41	81.60 ± 2.13	68.15 ± 1.78	96.23 ± 1.19	93.38 ± 1.43	**97.60 ± 0.66**	6.87*	100.00
90	91.40 ± 1.56	81.40 ± 2.58	68.45 ± 3.06	96.85 ± 1.70	94.20 ± 1.36	**97.75 ± 0.89**	6.95*	100.00

5. Experimental results and discussion

CMK outperforms our baseline kernel clearly in almost all training set percentages on SCIENCE dataset. This can be observed from Table 4. CMK demonstrates much better performance than linear kernel on this dataset, in all training set percentages except 5%. The performance gain is specifically obvious starting from 10% training set percentage. For instance at training set percentages 30%, 50%, 70%, 80% and 90% the accuracies of CMK are 95.07%, 96.71%, 97.12%, 97.6% and 97.75% while the accuracies of linear kernel are 86.73%, 88.94%, 90.58, 91.33% and 91.4%%; respectively. CMK also has better performance than our previous semantic kernels IHOSK, and HOTK at training set percentages between 30% and 90% as shown in Table 4. The highest gain of CMK over linear kernel on this dataset is at 30% training set percentage which is 9.62%. Also it should be noted that, there is a performance gain of CMK over linear kernel 5.41% at training set percentage 10%, which is of great importance since usually it is difficult and expensive to obtain labeled data in real world applications. Additionally, according to Table 4 we can conclude that the performance differences of CMK while passing from one training set percentage to another are compatible with the term coverage ratios at those training set percentages. For instance at training set percentage 30%, term coverage jumps to 98.01% from its previous value at 10% that is 82.28%. Similar behavior can be observed at performance of CMK while going through 10% training set percentage to 30% training set percentage; where it generates the accuracies 82.19% and 95.07%; respectively. This means an accuracy change of 12.88% between 10% and 30% training set percentages.

Also, at all training set percentages CMK has an absolute superiority than both polynomial kernel and RBF on SCIENCE dataset. Actually this superiority on polynomial and RBF remains the same at almost all the training set levels of all datasets in this study. This can be observed from the following experiment results tables.

Additional to CMK, that is calculated with Eqs. (36) and (37) we also built a second-order version of CMK with the name Second-Order Class Meaning Kernel (SO-CMK) with the following equation:

$$k_{SO-CMK}(d_1, d_2) = d_1(SS)(SS)d_2^T \tag{41}$$

where S is our term-by-term meaning matrix that is also used for CMK. Transformations are done with

$$\varphi(d_1) = d_1 SS \text{ and } \varphi(d_2) = SSd_2^T \tag{42}$$

where $\varphi(d_1)$ and $\varphi(d_1)$ are transformation functions of kernel from input space into feature space for the documents d_1 and d_2, respectively. In other words, here M is a semantic proximity matrix of terms and classes which shows semantic relations between terms. In this case semantic relation between two terms is composed of corresponding class based meaning values of these terms for all classes. So if these two terms are important terms in the same class then the resulting semantic relatedness value will be higher. In contrast to the other semantic kernels that makes use of WordNet or Wikipedia in an unsupervised fashion, CMK directly incorporates class information to the semantic kernel. Therefore, it can be considered as a supervised semantic kernel.

We also recorded and compared the total kernel computation time of our previous semantic kernels IHOSK and HOTK and CMK. All the experiments presented in this paper are carried on our experiment framework, Turkuaz, which directly uses WEKA (Hall et al., 2009) on a computer with two Intel(R) Xeon(R) CPUs at 2.66 GHz with 64 GB of memory. Our semantic kernel's computation time on each dataset is recorded in terms of seconds and they are proportionally converted into percentages by making the longest run time 100.According to this conversion, for instance on SCIENCE dataset; IHOSK (Altınel et al., 2014a), SO-CMK, CMK and HOTK (Altınel et al., 2014b) estimates the following time units in order;100, 55, 32, and 27, respectively, which is shown in Fig. 5.

These values are not surprising since the complexity and running time analysis supports them. In IHOSK (Altınel et al., 2014a), there is an iterative similarity calculation between documents and terms, which completes totally in four steps including corresponding matrix calculations as in shown in Eq. (13) and Eq. (14). As it is discussed in (Bisson and Hussain, 2008) producing the similarity matrix (SC_t) has overall complexity $O(tn^3)$ where t is the number of iterations and n is the number of training instances. Since in our experiments we fixed $t=2$ we obtain $O(2n^3)$ complexity. On the other hand HOTK (Altınel et al., 2014b) has complexity $O(n^3)$ as it can be noted from Eq. (16). CMK also has a complexity of $O(n^3)$ like HOTK, but additional to the calculations made for HOTK, CMK has a phase of calculating meaning values which makes CMK run slightly longer than HOTK as shown in Fig. 5. Moreover, SO-CMK includes additional matrix multiplications as a result it runs longer than CMK. Since the IHOSK involves much more matrix multiplications than both HOTK and the proposed work of the CMK, it runs almost three times longer than the proposed approach on a relatively small dataset with 2000 documents and 2000 attributes.

We also compare CMK with a kernel based on a similar method of TF-ICF which is explained in Section 2.3. We compare the results of TF-ICF to CMK with Eq. (29) which indeed a supervised approach as mentioned in Section 2.4. Additionally we also created an unsupervised version of Meaning kernel, Unsupervised Meaning Kernel (UMK), by using a single document as our context (the P value in Eq. (29)) instead of using a class of documents. This introduces an unsupervised behavior into CMK since our basic unit is not class but instead a single document. The results are shown in Fig. 6. The CMK has much better performance than both UMK and TF-ICF in almost all training set percentages except 10%. Starting from training set percentage 10% the difference between the performance of CMK and the other two algorithms start to increase.

According to our experiments, the CMK demonstrates a notable performance gain on the IMDB dataset, which can be seen in Table 5. The CMK outperforms our baseline, linear kernel, in all training set percentages also making a significant difference at training set percentage 30% based on Students t-Tests results. In training set percentage 30% the performance of the CMK is 90.54%

while the performance of linear kernel is only 85.57%. It is also very promising to see that the CMK is superior to both linear kernel and our previous algorithms IHOSK (Altınel et al., 2014a) and HOTK (Altınel et al., 2014b) throughout all training set percentages.

Table 6 presents the experiment results on the POLITICS dataset. In this dataset, the CMK's performance is higher than linear kernel's in all training set percentages except 5% and 10%. Furthermore, the CMK is performs better than both IHOSK and HOTK in almost all training set percentages except 5% and 10%. Only in training set percentages 5% and 10%, the IHOSK gives better accuracy than the CMK, but CMK still remains better than both polynomial kernel and RBF kernel at those training set percentages.

For COMP dataset, the CMK outperforms linear kernel in all training set percentages except 5% as shown in Table 7. The CMK yields higher accuracies compared to linear kernel, IHOSK and HOTK. The differences between CMK and linear kernel are statistically significant according to Student's *t*-test at training levels 10%, 30%, 50%, 70%, 80%, and 90%.

Experiment results on RELIGION dataset are presented in Table 8. These results show that the CMK has superiority starting from 30% training set percentage among all of the other kernels. For instance at training set percentage 30% CMK's gain over linear kernel is 8.58%. Also, in training set percentages 30% and 50%, the CMK shows a significant improvement over linear kernel.

Table 9 presents the experiment results on mini-news group dataset. According to these results the CMK outputs better accuracy than linear kernel at training set percentages 30%, 50%, 70%, 80% and 90%. But in overall the CMK is not as good as HOTK on this dataset, which can be explained by the capability of HOTK for capturing latent semantics between documents by using higher-order term co-occurrences as explained in Section 2.2. These latent relations may play an important role since the number of classes is relatively high and the number of documents per class is much smaller yielding a higher sparsity that can be observed from the term coverage statistics.

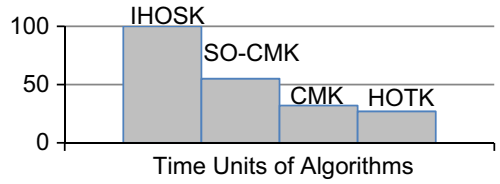

Fig. 5. The total kernel computation time units of IHOSK, SO-CMK, CMK and HOTK on SCIENCE dataset at 30% training set percentage.

Since some of the datasets used in this study are also used in (Ganiz et al., 2009), we have the opportunity to compare our results with HOSVM. For instance at training level 30%, on COMP dataset; 75.38%, 78.71%, 75.97%, and 84.31% accuracies are gathered by linear kernel, IHSOK, HOTK and CMK as mentioned in above tables and paragraphs. On the same training level HOSVM achieves 78% accuracy according to the Fig. 2(d) in (Ganiz et al., 2009). This comparison shows CMK outperforms HOSVM by approximately 8.28% gain. Actually CMK's superiority on HOSVM carries on other datasets such as RELIGION, SCIENCE and POLITICS. For instance on POLITICS dataset while HOSVM' performance is about 91%, CMK reaches 96.53% accuracy, which produces a gain of 8.95%. Very similar comparison results can be seen at a higher training level such as 50%.For example the experiment results of 88.94, 92, 94.97, 90.84, 96.71 are achieved by linear kernel, HOSVM, IHSOK, HOTK and CMK at SCIENCE dataset at training level 50%; respectively.

6. Conclusions and future work

We introduce a new semantic kernel for SVM called *Class Meanings Kernel* (CMK). The CMK is based on meaning values of terms in the context of classes in the training set. The meaning values are calculated according to the Helmholtz Principle which is mainly based on Gestalt theory and has previously been applied to several text mining problems including document summarization, and feature extraction (Balinsky et al., 2010, 2011a, 2011b, 2011c). Gestalt theory points out that meaningful features and interesting events appears in large deviations from randomness. The meaning calculations attempt to define meaningfulness of terms in text by using the human perceptual model of the Helmholtz principle from Gestalt Theory. In the context of text mining, the textual data consist of natural structures in the form of sentences, paragraphs, documents, topics and in our case classes of documents. In our semantic kernel setting, we compute meaning values of terms, obtained using the Helmholtz principle in the context of classes where these terms appear. We use these meaning values to smoothen document term vectors. As a result our approach can be considered as a supervised semantic smoothing kernel which makes use of the class information. This is one of the important novelties of our approach since the previous studies of semantic smoothing kernels does not incorporate class specific information.

Our experimental results show the promise of the CMK as a semantic smoothing kernel for SVM in the text classification domain. The CMK performs better than commonly used kernels in the literature such as linear kernel, polynomial kernel and RBF, in most

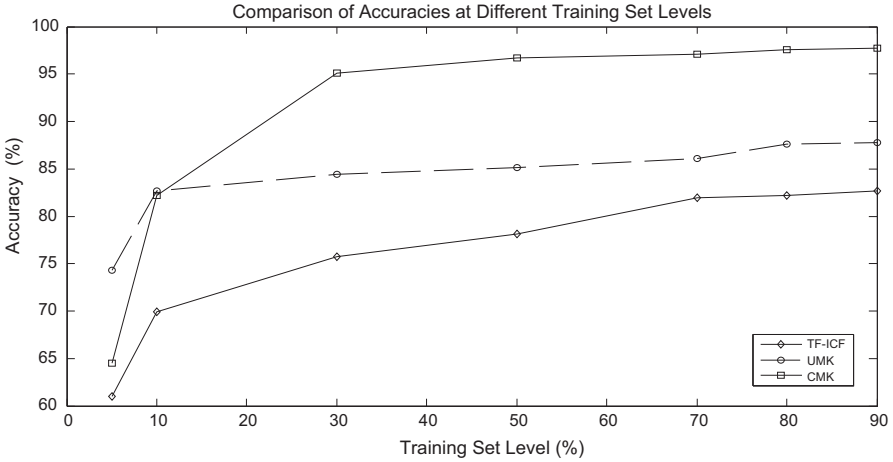

Fig. 6. Comparison of the accuracies of TF-ICF, UMK and CMK at different training set percentages on SCIENCE dataset.

Table 5
Accuracy of different kernels on IMDB dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	76.85 ± 1.31	69.20 ± 18.31	57.10 ± 28.93	76.98 ± 1.14	74.21 ± 0.24	**77.84 ± 2.99**	1.29	48.00
10	82.99 ± 1.76	64.56 ± 1.64	63.65 ± 2.69	82.55 ± 2.32	82.23 ± 0.42	**84.51 ± 1.45**	1.83	61.51
30	85.57 ± 1.65	74.65 ± 1.62	72.86 ± 1.76	87.16 ± 1.64	85.63 ± 1.69	**90.54 ± 0.65**	5.81*	86.35
50	88.46 ± 1.89	80.65 ± 0.89	78.06 ± 1.47	89.40 ± 1.91	87.20 ± 0.33	**92.30 ± 0.59**	4.34	95.91
70	89.93 ± 1.18	81.13 ± 0.83	80.44 ± 0.78	91.31 ± 0.87	90.41 ± 0.55	**93.23 ± 0.70**	3.67	99.17
80	90.65 ± 1.09	84.76 ± 0.34	81.07 ± 0.4	92.38 ± 1.43	91.37 ± 0.98	**93.43 ± 0.94**	3.07	99.71
90	91.75 ± 1.14	85.69 ± 1.22	82.16 ± 0.52	92.63 ± 1.19	91.59 ± 0.27	**93.65 ± 0.37**	2.07	99.98

Table 6
Accuracy of different kernels on POLITICS dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	79.01 ± 2.65	56.69 ± 6.79	55.74 ± 6.43	**82.27 ± 4.60**	80.72 ± 1.56	65.80 ± 3.99	−16.72	58.60
10	84.69 ± 1.24	62.45 ± 6.67	65.33 ± 3.96	**88.61 ± 2.10**	84.89 ± 2.15	78.50 ± 6.05	−7.31	75.02
30	92.04 ± 1.06	83.30 ± 4.57	80.34 ± 4.05	93.61 ± 1.08	88.31 ± 1.22	**95.03 ± 0.70**	3.25	96.37
50	93.73 ± 0.57	89.43 ± 2.03	87.95 ± 2.18	93.55 ± 3.58	90.29 ± 0.79	**96.43 ± 0.58**	2.88	99.43
70	94.55 ± 1.21	91.02 ± 1.50	87.84 ± 1.79	93.24 ± 3.08	90.15 ± 1.15	**95.82 ± 0.62**	1.34	99.97
80	94.03 ± 0.91	90.77 ± 1.50	88.50 ± 1.12	95.30 ± 1.82	92.50 ± 1.60	**96.73 ± 0.87**	2.87	100.00
90	94.86 ± 1.26	92.20 ± 1.81	89.80 ± 2.18	95.80 ± 2.28	92.46 ± 2.01	**96.53 ± 1.57**	1.76	100.00

Table 7
Accuracy of different kernels on COMP dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	56.75 ± 4.72	37.23 ± 3.57	35.26 ± 6.16	**68.12 ± 1.04**	60.22 ± 3.00	55.97 ± 5.01	−1.37	48.26
10	65.45 ± 2.77	44.36 ± 3.07	41.11 ± 5.51	**72.71 ± 0.43**	66.70 ± 1.14	70.21 ± 3.88	7.27*	65.19
30	75.38 ± 2.12	60.90 ± 3.00	48.16 ± 8.49	78.71 ± 0.04	75.97 ± 1.04	**84.31 ± 0.91**	11.85*	91.51
50	77.89 ± 1.60	64.60 ± 2.18	51.23 ± 5.88	82.18 ± 1.13	78.68 ± 0.71	**85.02 ± 0.72**	9.15*	98.92
70	79.63 ± 1.59	66.87 ± 2.25	58.93 ± 4.42	84.67 ± 2.83	80.97 ± 1.18	**85.60 ± 1.16**	7.50*	99.83
80	79.00 ± 2.25	65.70 ± 3.97	57.70 ± 4.13	85.81 ± 0.54	81.58 ± 1.85	**85.78 ± 1.42**	8.58*	99.98
90	81.40 ± 2.47	67.48 ± 2.29	58.80 ± 2.75	85.96 ± 0.69	81.32 ± 1.46	**86.00 ± 2.32**	5.65*	100.00

Table 8
Accuracy of different kernels on RELIGION dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	74.73 ± 2.47	52.52 ± 7.38	60.39 ± 8.04	**77.73 ± 2.47**	65.33 ± 1.70	58.98 ± 7.21	−21.08	41.80
10	80.98 ± 2.69	66.98 ± 4.57	73.01 ± 3.42	**81.19 ± 1.92**	72.10 ± 1.95	71.39 ± 7.57	−11.84	59.03
30	83.87 ± 0.78	77.10 ± 2.48	77.10 ± 3.51	84.85 ± 1.84	83.50 ± 1.58	**91.07 ± 1.39**	8.58*	88.18
50	88.39 ± 0.93	84.17 ± 2.53	82.69 ± 3.44	88.96 ± 2.30	86.19 ± 1.35	**93.04 ± 0.64**	5.26*	96.16
70	89.68 ± 1.41	86.36 ± 3.05	84.76 ± 2.78	90.62 ± 1.18	87.26 ± 0.31	**93.47 ± 1.23**	4.23	99.37
80	90.70 ± 1.12	87.37 ± 1.81	84.83 ± 2.94	91.00 ± 0.20	88.90 ± 0.24	**93.37 ± 1.68**	2.94	99.80
90	91.65 ± 1.63	89.33 ± 2.29	85.13 ± 3.30	91.70 ± 1.73	89.00 ± 2.37	**93.80 ± 2.18**	2.35	99.99

Table 9
Accuracy of different kernels on MINI-NEWSGROUP dataset with varying training set percentages.

TS%	Linear	Polynomial	RBF	IHOSK	HOTK	CMK	Gain	Term coverage
5	52.38 ± 5.53	41.21 ± 1.27	38.61 ± 3.18	**61.29 ± 1.03**	49.69 ± 5.64	48.89 ± 2.62	−6.66	34.90
10	59.85 ± 3.88	51.31 ± 2.37	50.21 ± 4.48	64.15 ± 0.54	**66.24 ± 3.81**	59.53 ± 2.49	−0.53	50.08
30	72.84 ± 3.56	68.33 ± 3.23	66.33 ± 4.13	75.51 ± 0.31	**81.82 ± 2.04**	74.24 ± 1.71	1.92	76.16
50	78.87 ± 2.94	70.12 ± 3.14	67.06 ± 3.34	79.24 ± 0.31	**85.54 ± 1.20**	79.65 ± 1.64	0.99	87.65
70	80.05 ± 1.96	75.80 ± 2.66	70.40 ± 1.26	79.73 ± 0.45	**87.28 ± 1.13**	80.23 ± 1.58	0.22	94.27
80	82.63 ± 1.36	76.83 ± 1.20	71.83 ± 2.10	83.05 ± 0.58	**88.15 ± 1.58**	83.53 ± 1.72	1.09	96.22
90	84.65 ± 2.48	77.55 ± 4.65	72.15 ± 2.35	85.38 ± 1.28	**88.10 ± 2.80**	85.64 ± 2.87	1.17	98.55

of our experiments. The CMK also outperforms other corpus-based semantic kernels such as IHOSK (Altınel et al., 2014a) and HOTK (Altınel et al., 2014b), in most of the datasets. Furthermore, the CMK forms a foundation that is open to several improvements. For instance, the CMK can easily be combined with other semantic kernels which smooth the document term vectors using term to term semantic relations, such as the ones using WordNet or Wikipedia.

As future work, we would like to analyze and shed light on how our approach implicitly captures semantic information in the context of a class when calculating the similarity between two

documents. We also plan to implement different class-based document or term similarities in supervised classification and further refine our approach.

Acknowledgment

This work is 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 TÜBİTAK. We would like to thank Peter Schüller from Marmara University for his valuable discussions and feedback on the manuscript. We also would like to thank Melike Tutkan for his help in meaning implementations. The co-author Murat Can Ganiz would like to thank Selim Akyokuş for the valuable discussions on the meaning calculations.

References

Alpaydın, E., 2004. Introduction to Machine Learning. MIT press, Cambridge, MA.
Altınel, B., Ganiz, M.C., Diri, B., 2013. A novel higher-order semantic kernel. In: Proceedings of the IEEE 10th International Conference on Electronics Computer and Computation (ICECCO), pp. 216 – 219.
Altınel, B., Ganiz, M.C., Diri, B., 2014a. A semantic kernel for text classification based on iterative higher-order relations between words and documents. In: Proceedings of the 13th International Conference on Artificial Intelligence and Soft Computing (ICAISC), Lecture Notes in Artificial Intelligence (LNAI),vol. 8467, pp. 505–517.
Altınel, B., Ganiz, M.C., Diri, B., 2014b. A simple semantic kernel approach for svm using higher-order paths. In: Proceedings of the IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pp. 431 – 435.
Balinsky, A., Balinsky, H., Simske, S., 2010. On the Helmholtz principle for documents processing. In: Proceedings of the 10th ACM Document Engineering (DocEng).
Balinsky, A., Balinsky, H., Simske, S., 2011a. On the Helmholtz principle for data mining. In: Proceedings of the Conference on Knowledge Discovery, Chengdu, China.
Balinsky, A., Balinsky, H., Simske, S., 2011b. Rapid change detection and text mining. In: Proceedings of the 2nd Conference on Mathematics in Defence (IMA), Defence Academy, UK.
Balinsky, H., Balinsky, A., Simske, S., 2011c. Document sentences as a small world. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2583 – 2588.
Basili, R., Cammisa, M., Moschitti, A., 2005. A semantic Kernel to classify texts with very few training examples. In Proceedings of the Workshop Learning in Web Search, 22nd International Conference on Machine Learning (ICML).
Bisson, G., Hussain, F., 2008. Chi-Sim: a new similarity measure for the co-clustering task. In: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications (ICMLA), pp. 211–217.
Bloehdorn, S., Basili, R., Cammisa, M., Moschitti, A., 2006. Semantic kernels for text classification based on topological measures of feature similarity. In: Proceedings of the Sixth International Conference on Data Mining (ICDM), pp. 808–812.
Bloehdorn, S., Moschitti, A., 2007. Combined syntactic and semantic kernels for text classification. In: Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007), Lecture Notes in Computer Science, vol. 4425. Springer, Rome, Italy, pp. 307-318.
Boser, B. E., Guyon, I. M., Vapnik,V. N., 1992. A training algorithm for optimal margin classifier. In: Proceedings of the 5th ACM Workshop on Computational Learning Theory, pp. 144–152.
Budanitsky, A., Hirst, G., 2006. Evaluating WordNet-based measures of lexical semantic relatedness 2006. J. Comput. Linguist. 32 (1), 13–47.
Dadachev, B., Balinsky, A., Balinsky, H., Simske, S., 2012. On the Helmholtz principle for data mining. In: Proceedings of the International Conference on Emerging Security Technologies (EST), pp. 99–102.
Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R., 1990. Indexing by latent semantic analysis. J. Am. Soc. Inform.n Sci. 41 (6), 391–407.
Desolneux, A., Moisan, L., Morel, J.-M., 2008. From gestalt theory to image analysis: a probabilistic approachInterdisciplinary Applied Mathematics, vol. 34. Springer.
Dumais, S., Platt, J., Heckerman, D., Sahami, M., 1998. Inductive learning algorithms and representations for text categorization. In: Proceedings of the Seventh International Conference on Information Retrieval and Knowledge Management (ACM-CIKM), Bethesda, US, pp. 148–155.
Ganiz, M.C., Lytkin, N.I., & Pottenger, W.M., 2009. Leveraging higher order dependencies between features for text classification. In: Proceedings of the Conference Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), pp. 375 – 390.
Ganiz, M.C., George, C., Pottenger, W.M., 2011. Higher-order Naive Bayes: a novel non-iid approach to text classification. IEEE Trans. Knowl. Data Eng. (TKDE) 23 (7), 1022–1034.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software. An Update; SIGKDD Explorations 11 (1).
Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. 13 (2), 415–425.
Joachims, T., 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Springer, Berlin Heidelberg, pp. 137–142.
Jones, K.S., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Doc.28.1, 11–21.
Kamber, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition Morgan Kaufmann, San Francisco.
Kandola, J., Shawe-Taylor, J., Cristianini, N., 2004. Learning semantic similarity. Adv. Neural Inform. Process. Syst., 15,657–664,2003.
Kleinberg, J., 2002. Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), vol. 7, issue 4, pp. 373 – 397.
Ko, Y., Seo, J., 2000. Automatic text categorization by unsupervised learning. In: Proceedings of the 18th conference on Computational linguistics-vol. 1. Association for Computational Linguistics.
Kontostathis, A., Pottenger, W.M., 2006. A framework for understanding LSI performance. J. Inform. Process. Manag., 56–73.
Lee, J.Ho, Kim, M.H., Lee, Y.J., 1993. Information retrieval based on conceptual distance in IS-A hierarchies. J. Doc. 49 (2), 188–207.
Lertnattee, V., Theeramunkong, T., 2004. Analysis of inverse class frequency in centroid-based text classification. In: Proceedings of the IEEE International Symposium on Communications and Information Technology, ISCIT, vol. 2.
Luo, Q., Chen, E., Xiong, H., 2011. A semantic term weighting scheme for text categorization. J. Expert Syst. Appl. 38, 12708–12716.
Mavroeidis, D., Tsatsaronis, G., Vazirgiannis, M., Theobald, M., Weikum, G., 2005. Word sense disambiguation for exploiting hierarchical thesauri in text classification, Knowledge Discovery in Databases: PKDD. Springer, Berlin Heidelberg, pp. 181–192.
Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K., 1993. Five Papers on WordNet Technical report. Cognitive Science Laboratory, Princeton University.
Nasir, J.A., Karim, A., Tsatsaronis, G., Varlamis, I., 2011. A Knowledge-Based Semantic Kernel For Text Classification, String Processing and Information Retrieval, 261 – 266. Springer, Berlin Heidelberg.
Nasir, J.A., Varlamis, I., Karim, A., Tsatsaronis, G., 2013. Semantic smoothing for text clustering. Knowl.-Based Syst. 54, 216–229.
Platt, J.C., 1998. Sequential minimal optimization: a fast algorithm for training support vector machines. In: Scholkopf, Burges, Smola (Eds.), Advances in Kernel Method: Support Vector Learning. MIT Press, Cambridge, MA, pp. 185–208.
Poyraz, M., Kilimic, Z.H., Ganiz, M.C., 2012. A novel semantic smoothing method based on higher-order paths for text classification. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 615 – 624.
Poyraz, M., Kilimic, Z.H., Ganiz, M.C., 2014. Higher-order smoothing: a novel semantic smoothing method for text classification. J. Comput. Sci. Technol. 29 (3), 376–391.
Robertson, S.E., 2004. Understanding inverse document frequency: on theoretical arguments for IDF. J. Doc. 60 (5), 503–520.
Salton, G., Yang, C.S., 1973. On the specification of term values in automatic indexing. J. Doc. 29 (4), 11–21.
Scott, S., Matwin, S., 1998. Text classification using WordNet hypernyms. In: Proceedings of the ACL Workshop on Usage of WordNet in Natural Language Processing Systems, pp. 45–52.
Siolas, G., d'Alché-Buc, F., 2000. Support vector machines based on a semantic kernel for text categorization. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, vol. 5, pp. 205 – 209.
Steinbach, M., Karypis, G., Kumar, V., 2000. A comparison of document clustering techniques. In: Proceedings of the KDD Workshop on Text Mining.
Tsatsaronis, G., Varlamis, I., Vazirgiannis, M., 2010. Text relatedness based on a word thesaurus. J. Artif. Intell. Res. 37, 1–39.
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer, New York.
Wang, P., Domeniconi, C., 2008. Building semantic kernels for text classification using Wikipedia. In: Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 713 – 721.
Wang, T., Rao, J., Hu, Q., 2014. Supervised word sense disambiguation using semantic diffusion kernel. Engineering Applications of Artificial Intelligence, 27. Elsevier, pp. 167–174.
Wittek, P., Tan, C., 2009. A kernel-based feature weighting for text classification. In: Proceedings of the IJCNN-09, IEEE International Joint Conference on Neural Networks, pp. 3373–3379.
Zhang, P.Y., 2013. A HowNet-based semantic relatedness kernel for text classification. Indones. J. Electr. Eng. (TELKOMNIKA) 11, 4.
Zhang, Z., Gentile, A.L., Ciravegna, F., 2012. Recent advances in methods of lexical semantic relatedness–a survey. Nat. Lang. Eng. 1 (1), 1–69.