Information Processing and Management 54 (2018) 1129–1153

Contents lists available at ScienceDirect

Information Processing and Management

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

Semantic text classification: A survey of past and recent advances

Berna Altınel*, Murat Can Ganiz

College of Engineering, Department of Computer Engineering, Marmara University, Turkey

ARTICLE INFO

Keywords:
Text classification
Semantic text classification
Knowledge-based systems
Corpus-based systems
Neural language models
Deep learning

ABSTRACT

Automatic text classification is the task of organizing documents into pre-determined classes, generally using machine learning algorithms. Generally speaking, it is one of the most important methods to organize and make use of the gigantic amounts of information that exist in unstructured textual format. Text classification is a widely studied research area of language processing and text mining. In traditional text classification, a document is represented as a bag of words where the words in other words terms are cut from their finer context i.e. their location in a sentence or in a document. Only the broader context of document is used with some type of term frequency information in the vector space. Consequently, semantics of words that can be inferred from the finer context of its location in a sentence and its relations with neighboring words are usually ignored. However, meaning of words, semantic connections between words, documents and even classes are obviously important since methods that capture semantics generally reach better classification performances. Several surveys have been published to analyze diverse approaches for the traditional text classification methods. Most of these surveys cover application of different semantic term relatedness methods in text classification up to a certain degree. However, they do not specifically target semantic text classification algorithms and their advantages over the traditional text classification. In order to fill this gap, we undertake a comprehensive discussion of semantic text classification vs. traditional text classification. This survey explores the past and recent advancements in semantic text classification and attempts to organize existing approaches under five fundamental categories; domain knowledge-based approaches, corpus-based approaches, deep learning based approaches, word/character sequence enhanced approaches and linguistic enriched approaches. Furthermore, this survey highlights the advantages of semantic text classification algorithms over the traditional text classification algorithms.

1. Introduction

1.1. Traditional text classification and its challenges

Text mining studies steadily gain importance in recent years due to the wide range of sources that produce enormous amounts of data, such as social networks, blogs/forums, web sites, e-mails, and online libraries publishing research papers. The growth of electronic textual data will no doubt continue to increase with new developments in technology such as speech to text engines and digital assistants or intelligent personal assistants. Automatically processing, organizing and handling this textual data is a fundamental problem. Text mining has several important applications like classification (i.e., supervised, unsupervised and semi-

* Corresponding author.
E-mail addresses: berna.altinel@marmara.edu.tr (B. Altınel), murat.ganiz@marmara.edu.tr (M.C. Ganiz).

https://doi.org/10.1016/j.ipm.2018.08.001
Received 4 August 2017; Received in revised form 28 July 2018; Accepted 6 August 2018
Available online 20 August 2018
0306-4573/

supervised classification), document filtering, summarization, and sentiment analysis/opinion classification. Natural Language Processing (NLP), Machine Learning (ML) and Data Mining (DM) methods work together to detect patterns from the different types of the documents and classify them in an automatic manner (Sebastiani, 2005).

A traditional method for representing documents is called Bag of Words (BOW). This representation technique only include information about the terms and their corresponding frequencies in a document independent of their locations in the sentence or document. It is also called the Vector Space Model (VSM) since each document is represented as a vector of term frequencies in the vocabulary. Each of these terms in the vocabulary denotes an independent (orthogonal) dimension in the vector space, which usually results in a very high dimensional document vectors with only a few of them taking a frequency value which in turn yields to high sparsity. Furthermore, this representation does not take into account semantic associations between words. For instance, two words written as a different sequence of characters constitute different orthogonal dimensions of this vector space although they may be synonymous. Additionally, order of these words in the sentences are completely lost in the BOW representation. This approach mainly emphasizes the existence of some form of frequency information of terms. The BOW methodology makes the representation of documents simpler by disregarding the following several different semantic and syntactic relations between words in natural language: Firstly, it disregards the multi-word expressions by separating them into independent terms. Secondly, it treats polysemous words (words with multiple meanings) as a single entity because the word is separated from its neighboring words that determine its sense. Thirdly, the BOW approach maps synonymous words into distinct terms (Salton & Yang, 1973).

A text classifier is expected to label textual documents with pre-determined classes with an obvious assumption that each class consist of similar documents, usually talking about a particular topic that is different from the topics of other classes. However, vector space demonstration of texts usually results in high dimensionality and consequently high sparsity. This is a big difficulty especially when there are numerous class labels but inadequate training data for each of them. Obtaining labeled quality data for training is usually very expensive in real world applications. Accordingly, an accurate text classifier should have the capability of using this semantic information.

1.2. Semantic text classification and its advantages over traditional text classification

In semantic text classification methods, semantic relations between words are considered in order to, generally, measure similarity between documents. The semantic approach focuses on meaning of the words and hidden semantic connections between words and consequently between documents. Advantages of semantic text classification over traditional text classification are listed as:

- Implicit or explicit relationship discovery between words.
- Extracting and using latent relationships between words and documents.
- Capability to generate representative keywords for the existing classes.
- Semantic understanding of text, which improves accuracy of classification.
- Ability to handle synonymy and polysemy in compare to traditional text classification algorithms since they utilize semantic relationships between words.

1.3. Overview of existing semantic text classification algorithms

In order to overcome the difficulties created by BOW feature representation as mentioned above, a number of semantic relatedness methods have been proposed to incorporate semantic relations between words in text classification. These methods can be grouped into five categories, namely; domain knowledge-based (ontology-based) methods, corpus-based methods, deep learning based methods, word/character enhanced methods and linguistic enriched methods (Fig. 1):

- *Domain knowledge-based (Language dependent) approaches*: An ontology or thesaurus is used by domain knowledge-based systems to identify concepts in documents. Examples of knowledge bases are dictionaries, thesauri and encyclopedic resources. Common knowledge bases are WordNet, Wiktionary and Wikipedia. Among them WordNet is by far the most used knowledge-base.
- *Corpus-based (Language independent) approaches*: Certain mathematical computations are performed in these systems for exposing latent similarities between words in the training corpus (Zhang, Gentile, & Ciravegna, 2012). One of the well-known corpus-based algorithms is Latent Semantics Analysis (LSA) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990).
- *Deep learning based approaches*: In recent years, especially since 2006, deep learning or hierarchical learning, has gained much attention in machine learning applications. Deep learning is a hybrid research area that is in the intersection of neural networks, graphical modeling, optimization, pattern recognition, and signal processing.
- *Word/character sequence enhanced approaches*: Word/character sequence enhanced systems treat words or characters as string sequences, which are taken out from documents by traditional string-matching techniques.
- *Linguistic enriched approaches*: These approaches use lexical and syntactic rules for extracting the noun phrases, entities and terminologies from a document to develop a representation of the document. Types of semantic algorithms for text classification are shown in Fig. 1.

Many studies in the scientific literature (Aas & Eikvil, 1999; Aggarwal & Zhai, 2012; Berry, 2004; Hotho, Nürnberger, & Paaß, 2005; Sebastiani, 2005) focus on traditional methods for text mining. Furthermore, there are also surveys that focus on particular type of classification algorithms such as kernel methods (Campbell, 2002; Jäkel, Schölkopf, & Wichmann, 2007). There are also

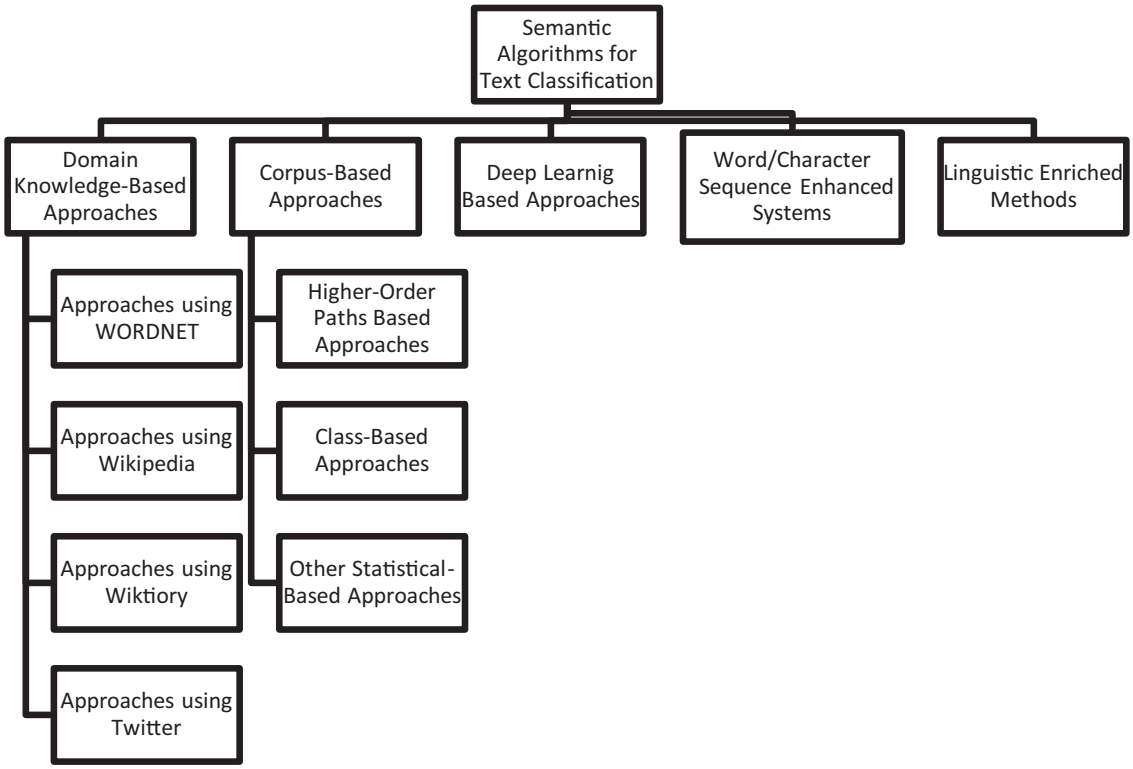


Fig. 1. Types of semantic approaches for text classification.

surveys about the techniques of semantic similarity measurement between words (Elavarasi, Akilandeswari, & Menaga, 2014; Soleimandarabi, Mirroshandel, & Sadr, 2015; Zhang et al., 2012). Moreover, there is a discussion about types of semantic relationships between words on the textual data of the social networks (Irfan et al., 2015). Similar to our topic, there are surveys on semantic document clustering such as Naik, Prajapati, and Dabhi (2015) and Saiyad, Prajapati, and Dabhi (2016). Nevertheless, there is a lack of comprehensive discussion on the analysis of different types of semantic text classification algorithms such as domain knowledge-based approaches, corpus-based approaches, deep learning based methods, word/character enhanced systems and linguistic enriched algorithms. In contrast to existing surveys, this survey strives to concentrate and address all the above-mentioned deficiencies by presenting a focused and deeply detailed literature review on the application of semantic text classification algorithms.

The rest of the survey is organized as follows: There will be a detailed description about knowledge-based resources and example domain knowledge-based studies in Section 2. After that, a detailed discussion about corpus-based studies including higher-order paths based studies, class-based studies and other statistical approaches will be given in Section 3. Section 4 summarizes deep learning algorithms. Following this, word/character sequences enhanced studies and linguistic enriched-based studies will be presented in Sections 5 and 6, respectively. A global comparison of top-performing algorithms of each type will be given in Section 7. Next, current challenges, the future directions for the researchers and the conclusion will be presented in Section 8.

2. Semantic text classification with knowledge-based approaches

2.1. Overview of the approach

These approaches take advantage of knowledge-based sources to enrich the text representation. WordNet, dictionaries, thesauri and encyclopedic resources are widely used examples of domain knowledge sources. Some examples of these studies are as follows: Kozima and Furugori (1993), Morris and Hirst (1991) and Jarmasz and Szpakowicz (2003). WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1993), Wiktionary[1] and Wikipedia[2] are the most commonly used general-purpose knowledge bases.

2.1.1. WordNet as a knowledge-base for text classification

WordNet is a lexicalized ontology of English words. It groups verbs, nouns, adjectives and adverbs into synsets, each expressing a

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

different concept. Therefore, after a search for a word in WordNet may bring several synsets related to different senses or concepts. There is a short definition for each word in WordNet. Most importantly, WordNet also provides the semantic relations between the words such as hypernym, meronym, synonymy and antonym.

WordNet has very limited coverage of vocabularies from specialized domains and proper nouns (Zhang et al., 2012). The GermaNet (Kunze & Lemnitzer, 2002) is German equivalent to WordNet and has been used in some applications (Zesch & Gurevych, 2010; Zesch, Muller, & Gurevych, 2008).

WordNet is one of the most commonly used domain knowledge sources for text classification. Some examples of WordNet-based methods are Agirre et al. (2009), Hughes and Ramage (2007), Leacock and Chodorow (1998), Li, Bandar, and McLean (2003), Wu and Palmer (1994), Zesch and Gurevych (2010), Zhang, Gentile, and Ciravegna (2011).

2.1.2. Wiktionary as a knowledge-base for text classification

Wiktionary is a multilingual free dictionary, which provides short definitions of each concept. Furthermore, each entry in Wiktionary is an article page related to a term and differentiates one or more word classes. Wiktionary also provides lexical semantic relations, which are accessible from WordNet such as hyponymy, synonymy, hypernym and antonym.

Wiktionary covers twice the amount of words than in WordNet as discussed in Meyer and Gurevych (2010). On the other hand, about half the amount of WordNet lexicons are missing in the English Wiktionary. Navarro et al. (2009) show that the lexical–semantic information is not always encoded for the words in the same word class.

Compared with WordNet, the use of Wiktionary in semantic relatedness is rarely investigated. One of them is Zesch et al. (2008) which try to adapt numerous WordNet-based methods to the Wiktionary semantic graph.

2.1.3. Wikipedia as a knowledge-base for text classification

Wikipedia is a multilingual encyclopedia created and maintained by collaborative effort (Zhang et al., 2012). Wikipedia articles are hyperlinked; which indicates some level of semantic relatedness between different concepts. Additionally, the articles are classified with usually more than one class labels. A more valuable semantic information lies in the 'redirect' mechanism in Wikipedia, which can be considered as groups of synonyms and aliases. Similarly, phrases and polysemous names are encoded in 'disambiguation' pages that list distinct meanings.

Wikipedia has many advantages over Wiktionary and WordNet. First, Wikipedia covers a considerable number of concepts, proper nouns and domain specific vocabularies. According to the analysis and the experimental results in Halavais and Lackaff (2008) the coverage of topic-specific knowledge is generally much better in Wikipedia.

2.2. Text classification studies of knowledge-based approaches

This section provides an overview of knowledge-based approaches offered for text classification; including their descriptions, experimental results and advantages/disadvantages.

Siolas and d'Alché-Buc (2000) propose a semantic kernel that is built by using the semantic relations of English words in WordNet. Semantic similarities between terms are calculated with the help of the hierarchies and links between words in WordNet. They use this information as a smoothing technique to enhance the Gaussian kernel and standard k-NN algorithm. According to their experimental results, using a semantic similarity metric increases the classification accuracy. For instance, the classification performance with semantic k-NN is 87.12% in a subgroup (talk.politics.gun) of 20NewsGroups[3] dataset while the classification performance of standard k-NN is just 70.05%.

Bloehdorn, Basili, Cammisa, and Moschitti (2006) present super concept declaration. They build a kernel function that gets the knowledge of topology through their super concept expansion. This kernel function is given in Eq. (1), where Q is a semantic smoothing matrix and it is composed of P and P^T, which include super-concept information about the corpus. According to the experimental results and analysis, their kernel function reaches significant improvement in classification performance where the feature demonstrations are highly sparse or little training data exists (Bloehdorn et al., 2006).

$$k(d_p, d_q) = d_p P P^T d_q^T \tag{1}$$

Bloehdorn and Moschitti (2007) offer an algorithm called as Semantic Syntactic Tree Kernel (SSTK). SSTK is comprised of syntactic dependencies like linguistic structures with semantic knowledge that is collected from WordNet. Luo, Chen, and Xiong (2011) use WordNet as a resource of semantic knowledge base.

Similarly, the study of Nasir, Karim, Tsatsaronis, and Varlamis (2011) uses WordNet to create a semantic proximity matrix with Omiotis (Tsatsaronis, Varlamis, & Vazirgiannis, 2010). Omiotis is a knowledge-based measure in order to compute the relatedness between terms. Nasir et al. (2011) integrated this information into a Term Frequency-Inverse Document Frequency (tf-idf) weighting technique. Their algorithm has higher classification accuracy than the standard BOW demonstration. They extend their study by considering just top-k semantically related words and performing experiments on larger text datasets (Nasir, Varlamis, Karim, & Tsatsaronis, 2013).

Wang and Domeniconi (2008) present a semantic proximity matrix which is composed of the following measures: (1) content-based similarity measure which is based on Wikipedia articles' BOW demonstration, (2) out-link category-based measure that gets an

[3] http://www.cs.cmu.edu/~textlearning.

information associated to the out-link categories of two correlated articles in Wikipedia, (3) distance measure that is computed from the length of the shortest path connecting the two categories of two articles belong to, in Wikipedia's category taxonomy. Their experimental results show that adding semantic information from Wikipedia into document representation overcomes some of the deficiencies of the BOW approach and increases the text classification accuracy.

Suganya and Gomathi (2013), Torunoğlu, Telseren, Sağtürk, and Ganiz (2013) and Yang, Li, Ding, Li (2013) also use Wikipedia as background knowledge in their classification frameworks.

Semantic Diffusion Kernel takes advantage of both exponential transformation and WordNet in order to build a kernel function (Kandola, Shawe-Taylor, & Cristianini, 2004; Wang, Rao, & Hu, 2014). According to the experiments in Wang et al. (2014) the diffusion matrix makes use of higher-order co-occurrences to acquire hidden semantic connections between terms in the Word Sense Disambiguation (WSD) tasks from SensEval.

Zhang, Yoshida, and Tang (2008) concentrate on using multi-word phrases for text representation by using the syntactical structure of noun phrases. They offer two strategies, namely; general concept representation and subtopic representation, to represent the documents. Their two representations include extracted multi-word phrases from the WordNet library (Zhang et al., 2008). Their first strategy uses the general concepts while the second strategy uses the subtopics of the general concepts for the representation of documents.

They report three advantages of using multi-word phrases (Zhang et al., 2008): (1) Using multi-word phrases decreases the number of dimensions. (2) Acquiring multi-word phrases is an easy task. (3) Multi-word phrases carry more semantics than individual terms. They report that their methodology with multi-word linear kernel is superior to the customary linear kernel (Zhang et al., 2008).

2.3. Performance comparison of knowledge-based approaches

Table 1 presents the knowledge-based approaches including their methodologies, experiment settings and experimental results.

For knowledge-based techniques, it is practically challenging to create the same experimental environment. However, it is possible to compare works that use the same dataset. For example, Siolas and d'Alché-Buc (2000) and Nasir et al. (2011) use the same 20NewsGroups dataset. Siolas and d'Alché-Buc (2000) get 88.52% classification accuracy with five experimental runs while Nasir et al. (2011) get 92.93% classification accuracy with 10 experimental runs. They both use WordNet as a thesaurus in order to calculate the semantic relatedness between terms. Furthermore, Nasir et al. (2011) use Omiotis measure (Tsatsaronis et al., 2010) in order to build the semantic proximity matrix and they integrate this measure into a tf-idf weighting methodology. On the other side, Siolas and d'Alché-Buc (2000) combine the proximity matrix from WordNet with the radial basis kernel. The performance improvement in the study of Nasir et al. (2011) may be due to the usage of a more extensive version of WordNet in comparison to the study in Siolas and d'Alché-Buc (2000) and the Omiotis measure (Tsatsaronis et al., 2010). Additionally, it should be noted from Table 1 that the classification accuracies in the studies that use Wikipedia as a knowledge base, are higher than the classification accuracies in the studies that use WordNet as a knowledge base. The performance enhancement in the study in Wang and Domeniconi (2008) may be caused by several advantages of Wikipedia over WordNet and Wiktionary. Most of all, there are many proper nouns, concepts and domain-specific vocabularies in Wikipedia's coverage. Moreover, Wikipedia has tremendous coverage of domain terminology and semantic relations that rivals a professional thesaurus as mentioned in Zhang et al. (2012). As it is discussed in Zhang et al. (2012), the denser relations between article pages and categories in Wikipedia, as well as more extensive content, also infer richer lexical semantic information. Furthermore, according to Table 1, Yang et al. (2013) get 93.00% classification accuracy in their study in which Wikipedia is used as background knowledge. They present a novel approach to combine lexical and semantic features for short text classification and put forward a new measure method to select lexical features from short texts. Experimental results indicate both the improvement of the feature selection and classification for short texts. Navigli and Ponzetto (2012) created BabelNet, which is a combination of WordNet and Wikipedia and can be further used for classification tasks as ontology.

3. Semantic text classification with corpus-based (language-independent) approaches

3.1. Overview of the approach

Corpus-based systems use statistical analysis in the set of training documents to discover hidden connections between them (Zhang et al., 2012). Corpus-based systems are also called language-independent systems since they are independent from any knowledge source such as WordNet and Wikipedia. This advantage ensures these systems do not need the processing of a large external knowledge base. Furthermore, being constructed from corpus-based statistics makes corpus-based systems up to date in the context of the corpus. Similarly, there is no coverage problem for these systems as the semantic relations between terms are specific to the domain of the corpus. Besides all these advantages, corpus-based systems can easily become knowledge-based systems by either combining with any ontology/thesaurus or deriving a semantic and syntactic structure by their own corpus. For instance, Harrington (2010) and Wojtinnek and Pulman (2011) have built a semantic and syntactic structure from a corpus with the help of NLP techniques such as Named Entity Recognition (NER) and syntactic parsing. Harrington (2010) and Wojtinnek and Pulman (2011) state that corpus-based systems may provide better coverage of domain-specific information than knowledge-based systems. However, pre-processing a large corpus of texts brings substantial computational cost, which is a major subject to be considered with corpus-based systems (Pantel, Crestan, Borkovsky, Popescu, & Vyas, 2009).

Table 1
Performance comparison of domain knowledge-based approaches across different datasets.[a]

Author	Year	Ontology or Thesaurus	Approach	Dataset	Performance metric	Results
Scott and Matwin	1998	WordNet	Supervised (Ripper Algorithm with Hypernym density representation)	USENET2[b] bionet.microbiology bionet.neuroscience	Average accuracy (10-fold) cross-validation	63.57
Rodriguez et al.	2000	WordNet	Supervised (Rocchio and Widrow-Hoff algorithms)	Reuters-21578 TC test collection (with 93 categories)	Precision, recall	50.2
Siolas and d'Alché-Buc	2000	WordNet	kNN	Subgroups of 20NewsGroups	Average accuracy (5- experimental runs)	80.13
Siolas ans d'Alché-Buc	2000	WordNet	Supervised (SVM)	20NewsGroups	Average accuracy (5 experimental runs)	88.52
Bloehdorn et al.	2006	WordNet	Supervised (SVMlight[c])	Reuters-21578 (%5 subset of the dataset)	Absolute macro F_1 scores, (10 experimental runs)	62.00
Bloehdorn and Moschitti	2007	WordNet	Supervised SVM-light-TK[d]	TRECQA[e]	macro F_1 scores	70.00
Gabrilovich and Markovitch	2007	Wikipedia	Explicit Semantic Analysis	Australian Broadcasting Corporation's news mail service (Lee, Pincombe, & Welsh, 2005)	classification accuracy	72.00
Wang and Domeniconi	2008	Wikipedia	Supervised (LIBSVM[f])	20NewsGroups	Micro-averaged precision results	89.92
Zhang et al.	2008	WordNet	Supervised (Multi-words with strategy and Linear kernel)	Reuters-21578 (with the categories, 'grain", "crude", "trade" and "interest")	Average Accuracy (3 fold cross validation)	80.77
Nasir et al.	2011	WordNet	Supervised SVM	20NewsGroups	Average Accuracy (10-fold) cross-validation)	92.93
Zhang et al.	2012	WordNet	Supervised Semantic Diffusion Kernel	The line data 2 The hard data 1 The serve data 578	Classification results(micro-F1)	i.) Line dataset: 84.09 ii.) Hard dataset: 84.84 iii.) Serve dataset: 87.00
Yang et al.	2013	Wikipedia	Supervised SVM	Ohsumed[g]	With 350 features	93.00
Suganya and Gomathi	2013	Wikipedia	Supervised Multilayer SVM + kNN	20Newsgroups	With 100 test records	93.00
Torunoğlu et al.	2013	Twitter, Wikipedia	NB	Twitter Sentiment 140 dataset (Twitter enriched with Wikipedia article titles) (Go, Bhayani, & Huang, 2009)	Average classification accuracy (10-fold) cross-validation), Training %70	74.00
Wang et al.	2014	WordNet	Semantic Diffusion Kernel	Line, hard, interest and serve data (Leacock, Towell, & Voorhees, 1993)	Macro, Micro F_1 scores	Micro F1 scores: i.) Line:84.09 ii.) Hard:84.94 iii.) Interest: 87.32 iv.) serve:87.00 Macro F1 scores: i.) Line:76.03 ii.) Hard:34.42 iii.) Interest: 72.82 iv.) serve:57.43

[a] The results may vary, even if the same data set is used because different preprocessing methods are used.
[b] http://www.liaad.up.pt/kdus/products/datasets-for-concept-drift.
[c] http://www.aifb.uni-karlsruhe.de/WBS/sbl/software/semkernel/.
[d] http://ai-nlp.info.uniroma2.it/moschitti/.
[e] http://l2r.cs.uiuc.edu/cogcomp/Data/QA/QC/.
[f] http://www.csie.ntu.edu.tw/&cjlin/libsvm.
[g] http://disi.unitn.it/moschitti/corpora.htm.

3.2. Text classification studies of corpus-based approaches

This section provides an overview of corpus-based approaches offered for text classification; including their descriptions, experimental results and advantages/disadvantages.

Several corpus-based studies in the literature utilize higher-order paths. A higher-order path can be considered as a chain of co-occurrences of entities (i.e., terms) in different records (i.e., documents). Kontostathis and Pottenger (2006) verify and demonstrate mathematically that Latent Semantic Indexing (LSI) (Deerwester et al., 1990), a well-known semantic algorithm, utilizes higher-order relations. The advantages of using higher-order paths between documents and terms are demonstrated in Fig. 2. There are three documents, d_1, d_2, and d_3, which contain a set of terms $\{t_1, t_2\}$, $\{t_2, t_3, t_4\}$, and $\{t_4, t_5\}$, respectively. The similarity value between documents d_1 and d_3 is zero arithmetically by using a customary similarity measure (e.g., dot product), which depends on just the number of shared terms. This simple example shows that the common terms between documents should not be the only factor when computing the similarity value between these documents. Because documents d_1 and d_3 have some connections in the dataset over d_2 as it can be observed in Fig. 2. This supports the idea that it is likely to obtain a non-zero similarity value between d_1 and d_3 by taking advantage of higher-order paths, which is not possible in the traditional BOW representation. In Fig. 2, there is also a higher-order path between t_1 and t_3. The similarity value through higher-order paths between any documents increases directly proportional to the number of connecting paths. Detecting higher-order paths is critical when two documents are written on the same topic using different but semantically closer sets of terms, which is a very common case in real world situations.

There are numerous LSI-based algorithms. For example, Zelikovitz and Hirsh (2004) present an LSI-based *k*-Nearest Neighborhood (LSI *k*-NN) algorithm in a semi-supervised setting for short text classification. In this work, the authors use the *k*-NN algorithm that depends on calculating similarities or distance between training documents and a test document in the transformed LSI space. A similar approach is used in a supervised setting in Ganiz, George, and Pottenger (2011).

Based on the work of Kontostathis and Pottenger (2006), Ganiz, Lytkin, and Pottenger (2009) and Ganiz et al. (2011) built a graph-based data representation inspired by their prior work (Ganiz, Kanitkar, Chuah, & Pottenger, 2006). A new Bayesian classification framework called Higher-Order Naive Bayes (HONB) is presented in Ganiz et al. (2009, 2011). HONB make use of implicit semantic relations between terms across documents by using higher-order paths. Higher-order paths are incorporated into a Bayesian learning framework based on the number of higher-order paths (Ganiz et al., 2009).

Higher-Order framework is built by implementing an original data-driven space transformation which lets vector space classifiers to utilize relational dependencies caught by higher-order paths between features as discussed in Ganiz et al. (2009) and Poyraz, Kilimci, and Ganiz (2014). This has led to the implementation of Higher-Order Support Vector Machines (HOSVM) (Ganiz et al., 2009). The higher-order learning framework depends on statistical relational methodology. It contains several supervised and unsupervised machine learning approaches in which relationships between different samples are leveraged with the help of higher-order paths (Edwards & Pottenger, 2011; Li, Wu, & Pottenger, 2005; Lytkin, 2009).

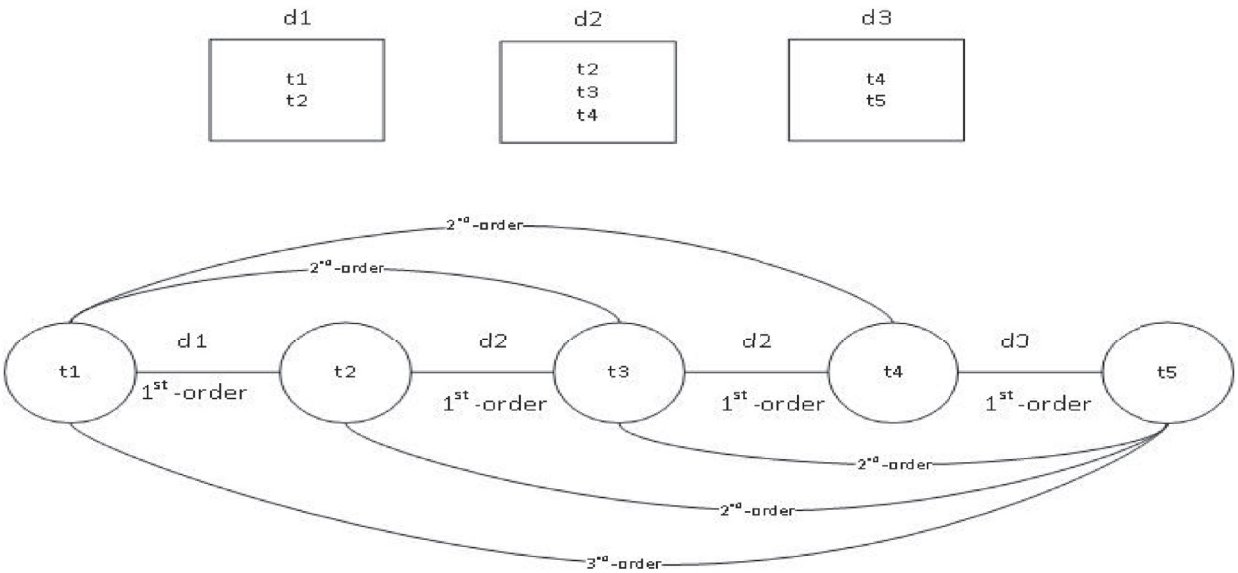
In another recent study related to higher-order term co-occurrence paths, an original semantic smoothing method, called Higher-Order Smoothing (HOS) for the Naive Bayes algorithm is presented in Poyraz, Kilimci, and Ganiz (2012) and Poyraz et al. (2014). Poyraz et al. (2012, 2014) use the relationships between instances of different classes in order to advance the parameter estimation when there is not enough labeled data. In order to achieve this, firstly nominal class attributes are converted to a number of binary attributes each representing a class label, which are added as columns to their document by term matrix. According to the extensive experimental results, authors show that HOS achieves significant classification accuracy improvements on several benchmark datasets.

In the work of Altınel, Ganiz, and Diri (2014a), a kernel method based on higher-order paths between texts and terms, known as Iterative Higher-Order Semantic Kernel (IHOSK), is presented. The similarity calculation in IHOSK is motivated by the similarity measure built in Bisson and Hussain (2008). According to this study, terms similarity matrix (*SC*) and document similarity matrix (*SR*) are produced iteratively.

In a novel text classification algorithm named Supervised Meaning Classifier (SMC) (Ganiz, Tutkan, & Akyokuş, 2015), the authors utilize the meaning values of terms. According to the study of Balinsky, Balinsky, and Simske (2011a), the meaning value of a term (word) w in a class c_j is computed with equation below:

$$meaning(w, c_j) = -\frac{1}{m}\log\left(\frac{k}{m}\right) - [(m-1)\log N] \tag{2}$$

where w shows a term, m denotes the occurrence of term w in class c_j, k specifies the frequency of term w in the whole dataset. $N = L/B$; L signifies the length of the dataset and B represents the length of the class c_j in terms (Balinsky, Balinsky, & Simske, 2011b). If a word's meaning score in a specific class is larger, then this means that this word is more informative for that class. Ganiz et al. (2015) calculate meaning values of the terms in a particular document for a particular class and sum them to obtain a relative class membership value of the document. In other words, that class membership of a particular document is determined by the sum of the meaning or the importance of its terms for that particular class. This is somewhat similar to the Naive Bayes algorithm where the class conditional document probability $P(D|C)$ is calculated by multiplying probabilities of the class conditional term probabilities $P(w|C)$ in addition to a class prior probability $P(C)$. The SMC classifier uses meaning calculations as explained in Section 3.2.1 in Ganiz et al. (2015) calculate the meaning scores of each word in the training set for each class, which constitutes the training phase. In the classification phase, for an unlabeled new test document, meaning scores of the words for a particular class are summed up to obtain class membership value. The class with largest membership value is chosen as the label of the instance. The SMC is shown to be



1st-order term co-occurrences: {t1, t2}, {t2, t3}, {t3, t4}, {t2, t4}, {t4, t5}

2nd-order term co-occurrences: {t1, t3}, {t1, t4}, {t2, t5}, {t3, t5}

3rd-order term co-occurrences: {t1, t5}

Fig. 2. Higher-order paths between terms through documents (Adapted from Altınel et al., 2015a).

superior to Multinomial Naive Bayes (MNB) and SVM with linear kernel, especially on inadequate training data (Ganiz et al., 2015).

Another method in this category is Class Meanings Kernel (CMK) (Altınel et al., Ganiz, & Diri, 2015a). Clearly, the main feature of this system is using the meaning calculation in kernel building process, for revealing semantic similarities between words and documents by smoothing the representation of the text documents. Although meaning calculation has been used in numerous fields, this work is the first to apply this technique to kernel function in the literature. Altınel et al. (2015a) generate the class-based term meaning matrix M using meaning calculations given in Eq. (2) to enrich the documents with semantic information for similarity calculations. The M matrix shows the meaningfulness of the words in each class. The authors compute S matrix, which extracts class, based semantic relations between terms as in Eq. (3). Precisely, S is a symmetric term-by-term matrix and the i, j element of S shows the semantic relatedness between words t_i and t_j.

$$S = \mathrm{M}\mathrm{M}^T \tag{3}$$

A related methodology is Class Weighting Kernel (CWK), which utilizes class-based term weights in a semantic kernel (Altınel, Diri, & Ganiz, 2015b). Biricik, Diri, and Sönmez (2009, 2012) motivate term weighs used in this study:

$$W_{w,c} = \log(\mathrm{tfc}_{w,c} + 1) \times \log\left(\frac{N}{N_w}\right) \tag{4}$$

where $tfc_{w,c}$ represents the total term frequency of term w in the documents of class c, N denotes the total number of documents in the corpus and N_w represents the total number of documents which have term w.

S matrix in Eq. (5) is built using the class-based term weighting approach to advance the BOW representation with semantic information (Altınel et al., 2015b).

$$S = WW^T \tag{5}$$

where W denotes a class-based term weighting matrix that is calculated with Eq. (4). In detail, the i, j element of S shows the semantic relatedness between terms t_i and t_j. S is a semantic smoothing matrix to transform texts from the input space to the feature space.

The information is stored in kernel matrix or Gram matrices in Wang et al. (2014) for SVM. The Gram matrix is given by:

$$G_{p,q} = k_{\mathrm{CWK}}(d_p, d_q) \tag{6}$$

The Gram matrix must satisfy the Mercer's conditions (i.e., being positive semi-definite) in order to be a valid kernel function (Alpaydın, 2004).

Liu, Chen, Zhang, Ma, and Wu (2004) offer a new method, namely, Local Relevancy Weighted Latent Semantic Indexing (LRW-LSI) to increase the classification performance of text classification. This method is different from Local LSI in that the documents in the local region are introduced using a smooth descending curve. This results in more related documents to the topic being assigned

higher weights. Therefore, it will be easier to concentrate on the semantic information that is actually most significant for the classification task. The experimental results show that LRW-LSI can improve the classification performance greatly using a much smaller dimensions compared to the global LSI and local LSI methods (Liu et al., 2004). They also perform a comparative study on global LSI and a few local LSI methods for text classification. Their experimental results show that local space is more appropriate for LSI than the global space. Although the global LSI can optimize demonstration of the completely original data in a low dimensional space, it does not help to improve the discrimination power of document classes; consequently, it always reduces the classification performance compared to the original term vector on classification (Liu et al., 2004). On the other hand, local LSI detects the significant local structure, which is critical in separating related documents from nearby unrelated documents. Thus, local LSI generates better classification performance than global LSI (Liu et al., 2004).

Bai, Padman, and Airoldi (2004) present a two-stage Bayesian algorithm that has capability to find dependencies among words. It also finds a vocabulary that is efficient for extracting sentiments. Their algorithm is able to capture dependencies among words and is able to find a minimal vocabulary. This minimal vocabulary needs to be sufficient for classification purposes. Their algorithm, namely Markov Blanket Classifier (MBC), has two stages: (1) It learns conditional dependencies among the words and encodes them into a Markov Blanket Directed Acyclic Graph (MB DAG) for the sentiment variable. (2) It uses a Tabu Search (TS) meta-heuristic strategy to fine-tune the MB DAG for getting a higher accuracy. They perform experiments on the Internet Movie Database (IMDB[4]) dataset and reach a cross-validated accuracy of 87.5% and AUC of 96.85% respectively.

Zhou, Zhang, and Hu (2008), suggest a novel semantic smoothing algorithm to solve the sparsity problem. Their method is based on extracting explicit topic signatures (e.g. words, multiword phrases) from a document and then statistically mapping them into single-word features. They conduct experiments on OHSUMED, LATimes, and 20NewsGroups to compare their semantic smoothing method with others. According to their experimental results, when the size of training set is small, the Bayesian classifier with semantic smoothing is superior to the classifiers with background smoothing, Laplacian smoothing, active learning classifiers and SVM classifiers.

In a more recent study, Kim et al. (2014) present language independent semantic (LIS) kernel that has the capability of computing the similarity between short-text documents without using grammatical tags and lexical databases. LIS kernel is composed of three parts: (1) Pattern extraction: It extracts patterns from a document by considering its syntactic information based on syntactic parse tree. (2) Semantic annotation: Extracted patterns are annotated in three annotation levels; word, document, and category. (3) Similarity computation: The similarity between two documents is calculated by using the extracted patterns based on the three annotation levels in the LIS kernel. A pattern kernel can be considered as a weighted linear combination of three types of kernels that are for the similarity computation depending on the related levels of semantic annotation. Then, the LIS kernel is used in text classification experiments on English and Korean datasets (Kim et al., 2014). The experimental results show that the LIS kernel has better performance compared to several existing kernels.

In the work of Uysal and Gunal (2014); a Genetic Algorithm (GA) is presented for text classification which is called oriented latent semantic features (GALSF). GALSF consists of two stages: feature selection and feature transformation. The state-of-the-art filtering-based methods are used in the feature selection stage. LSI with GA is used in the feature transformation stage, which results in better projection. The Effectiveness of the proposed method is comparatively evaluated on Enron1, Ohsumed and Reuters-21,578. For all datasets, the classification performance of GALSF is higher than the classification performance of other baselines in almost all test cases.

3.3. Performance comparison of corpus-based approaches

Table 2 presents the corpus-based approaches including their methodologies, experiment settings and experimental results.

Uysal and Gunal (2014) presents a GA that is oriented with latent semantic features. According to the experimental results in Uysal and Gunal (2014) and Zhou et al. (2008), Uysal and Gunal's study generates higher Micro-F1 in compare to the study in Liu et al. (2004) 1% training samples on OHSUMED dataset as shown in Table 2. On the other hand, the methodology in Liu et al. (2004) seems to results higher Micro-F1 than the methodology in Uysal and Gunal (2014) at 5% training set on Reuters-21,578 dataset according to Table 2. The performance difference may be due to the number of features and the methodology of capturing semantic information between words and documents. According to Table 2, Bai et al. (2004) reach 87.5% classification accuracy with their two-stage Markov Blanket Classifier. Additionally, the study in Kim et al. (2014) gains improvements over their baselines as 9.03% for BOW kernel (i.e., Base-line-1), 33.6% for ST kernel (i.e., baseline-2) and 23.66% for string kernel (i.e., baseline-3); respectively, as reported in Table 2.

Higher-order based studies reach noticeable classification performance on 20NewsGroups dataset based on the experimental results listed in Table 2. The reported classification results of higher-order based studies are generated as the average of 10 random trials on 20NewsGroups dataset at 5% training set level. According to experimental results in Ganiz et al. (2009, 2011), HONB outperforms both NB and SVM at 5% training set percentage not only on 20NewsGroups-Comp dataset but also on 20NewsGroups-Science, 20NewsGroups-Politics and 20NewsGroups-Religion datasets. Moreover, HONB seems superior to NB and SVM on these datasets at all training set percentages between 5% and 90% as shown in Ganiz et al. (2009, 2011). The experimental results on 20Newsgroups-Politics, 20Newsgroups-Science, 20Newsgroups-Religion and 20Newsgroups-Comp datasets show that leveraging higher-order term co-occurrence relations provides important enhancements in classification accuracies of both Bayesian and SVM-

[4] http://www.imdb.com/interfaces.

Table 2
Performance comparison of corpus-based approaches across different datasets.[a]

Author	Year	Approach	Dataset	Performance metric	Results
Liu et al.	2004	Local Relevancy Weighted LSI (SVM is used as classification algorithm)	Reuters-21578[b] with the most frequent 25 topics and used "Lewis" split which results in 6314 training examples and 2451 testing examples	Micro-F1(with 5% training samples, with 150 features)	94.00
Bai et al.	2004	two-stage Markov Blanket Classifier (MBC)	IMDB	Average accuracy (5-fold cross-validation)	87.5
Zhou et al.	2008	a novel semantic smoothing method for NB	Ohsumed, LATimes, and 20NewsGroups	Micro-F1, with 1% training samples	Ohsumed:41.3 LATimes:58.1 20NewsGroups:61.3
Ganiz et al.	2009	Higher-Order SVM (HOSVM)	20NewsGroups	Average accuracy of 10-random trial 5% training set percentage	Religion: 72.30 Science: 79.30 Politics: 79.20 Comp: 61.40
Ganiz et al.	2009, 2011	Higher-Order NB (HONB)	Subgroups of 20NewsGroups,20NewsGroups	Average accuracy of 10-random trial 5% training set percentage	20NewsGroups:64.65 Subgroups of 20NewsGroups: Religion: 74.18 Science: 84.32 Politics: 83.34 Comp: 65.06
Poyraz et al.	2012, 2014	Multivariate Bernoulli NB with Higher-Order Smoothing (MVNB + HOS) Multinomial NB with Higher-Order Smoothing (MNB + HOS)	20NewsGroups	Average accuracy of 10-random trial 5% training set percentage	MVNB + HOS:65.81 MNB + HOS:73:15
Altınel et al.	2014a	Iterative Higher-Order Semantic Kernel (IHOSK)	Subgroups of 20NewsGroups	Average accuracy of 10-random trial 5% training set percentage	Religion: 71.13 Science: 84.15 Politics: 82.27 Comp: 62.27
Altınel et al.	2014b	Higher-Order Term Kernel (HOTK)	Subgroups of 20NewsGroups	Average accuracy of 10-random trial 5% training set percentage	Religion: 63.24 Science: 76.63 Politics: 80.72 Comp: 60.22
Kim et al.	2014	Supervised (language independent (LIS) kernel based on semantic annotation)	open directory project (ODP), 8 Daum directory[c]	Average accuracy (30 repeated tests in each case)	Improvements over baselines: Baseline-1(BOW kernel):9.03%, Baseline-2(ST kernel): 33.66%, Baseline-3(String kernel): 23.66%
Uysal & Gunal	2014	GA oriented latent semantic features (GALSF)	Ohsumed, Reuters-21578	Micro-F1 (with 1% and 5% training samples)	1). 1% training samples: i). Reuters-21578 dataset:87.4 ii). Ohsumed dataset:58.9 2). 5% training samples: i). Reuters-21578 dataset:88.7 ii). Ohsumed dataset:62.4
Ganiz et al.	2015	Supervised Meaning Classifier (SMC)	Mini-newsgroups	Average accuracy 5% training set percentage	64.35
Altınel et al.	2015a	Class Meanings Kernel (CMK)	Subgroups of 20NewsGroups	Average accuracy 5% training set percentage	Religion: 58.98 Science: 64.51 Politics: 65.80 Comp: 55.97
Altınel et al.	2015b	Class Weighting Kernel (CWK)	Subgroups of 20NewsGroups	Average accuracy 5% training set percentage	Religion: 75.32 Science: 84.31 Politics: 83.49 Comp: 67.26

[a] The results may vary, even if the same data set is used because different preprocessing methods are used.
[b] http://www.daviddlewis.com/resources/testcollections.
[c] http://directory.search.daum.net.

based approaches. The improvements of HONB over NB and of HOSVM over SVM on those datasets are statistically significant at 5% training set percentage. HONB performs particularly well on the 20Newsgroups data, outperforming NB by 11.7% and SVM by 5.8% on average. HONB is also superior to HOSVM on all the datasets. In general, HOSVM outperforms SVM on all datasets.

According to the analysis in Poyraz et al. (2012, 2014), HOS is superior to all other classifiers such as Multivariate Binary NB (MVNB) (with default Laplace smoothing), NB with Jelinek-Mercer smoothing (MVNB + JM) and HONB by a wide margin in almost all training set levels.

Ganiz et al. (2015) present the accuracies of SMC, MNB and SVM at different training set levels on Mini-newsgroups dataset. According to the experimental results, SMC is superior to MNB, SVM at all training set sizes. The performance difference is especially noticeable at small training set levels.

Experimental results of IHOSK, HOTK and class weighting based algorithms (CMK and CWK) on subgroups of 20Newsgroups dataset are also presented in Table 2. According to the studies in Altınel et al. (2014a) and Altınel, Ganiz, & Diri (2014b), IHOSK and HOTK, are superior to all of the baseline kernels, linear kernel, polynomial and RBF at all training set percentages. IHOSK reaches higher classification accuracies than HOTK. This might result from the fact that IHOSK uses the higher-order relationships between both terms and documents while HOTK uses higher-order relations only between terms. According to Table 2, at training set level 5% the classification accuracies of CMK and CWK are 64.51% and 84.31% on 20Newsgroups-Science dataset; respectively. Additionally, CWK is superior to IHOSK, and HOTK at all training set levels as reported in Altınel et al. (2014b).

4. Deep learning approaches in text classification

4.1. Overview of the approach

Very recently, Deep Learning (DL) algorithms have been shown to be highly effective for analysis of textual documents. Traditionally, in machine learning, feature extraction is performed manually, which requires engineering skill and domain expertise. To avoid this costly and time consuming stage, good features for the learning task in hand can be learned automatically in the early layers of the DL algorithms. This is one of the cardinal advantages and the distinguishing character of DL algorithms. Human experts do not design features but they are learned from very large amounts of training data using a general-purpose learning procedure (LeCun, Bengio, & Hinton, 2015).

DL methods use different layers by non-linear connections to give different levels of representation of raw data (Najafabadi et al., 2015). Each layer applies a nonlinear transformation on its input and provides a representation in its output, which will be an input to the next layer in its hierarchical architecture. The objective is to learn the data in a hierarchical manner by passing the data through multiple transformation layers. It has been used by different online social networks applications such as sentiment analysis of short texts (Dos Santos & Gatti, 2014; Severyn & Moschitti, 2015b), recommender systems (Wang, Wang, & Yeung, 2015), and predicting the popularity (He et al., 2016; Tang, Qin, Liu, & Yang, 2015).

Before the popularity of deep learning (LeCun et al., 2015), most machine learning and signal processing architectures have a single linear or nonlinear feature transformation layer such as Gaussian mixture models (GMMs), hidden Markov models (HMMs), maximum entropy (MaxEnt) models, SVMs, and regression models. In this context, single layer architecture means that there is only one layer responsible for transforming the original input data points into a feature space.

Each layer applies a nonlinear transformation (i.e., it tries to extract essential explanatory factors in the data) on its input and provides a representation in its output which will be an input to the next layer in its layered architecture. The objective is to learn the data in a hierarchical manner by passing the data through multiple transformation layers.

Deep Learning algorithms are reasonably advantageous especially for unsupervised or semi-supervised learning where there is huge amount of unlabeled data, and typically learn data representations in a greedy layer-wise fashion (Bengio, Lamblin, Popovici, & Larochelle, 2007; Hinton, Osindero, & The, 2006). Empirical studies have demonstrated that deep learning often generate higher classification accuracy than traditional machine learning algorithms in many domains like speech recognition, computer vision, lately NLP and bioinformatics (Larochelle, Bengio, Louradour, & Lamblin, 2009). For more information, there are a number of literature surveys about deep learning in Deng (2014), Deng and Yu (2014), Hu, Zuo, Wang, and Liu (2016), LeCun et al. (2015) and Schmidhuber (2015).

4.2. Text classification studies of deep learning approaches

This section provides an overview of deep learning approaches offered for text classification; including their descriptions, experimental results and advantages/disadvantages.

There are two fundamental building blocks, unsupervised single layer learning algorithms that are used to construct deeper models: Auto-encoders and Restricted Boltzmann Machines (RBMs). Auto-encoders (Hinton & Zemel, 1994) are deep learning networks constructed of three layers: input, hidden and output. Auto-encoders try to learn the representations of the input in the hidden layer, and then it tries to reconstruct the input in the output layer based on these intermediate representations. A basic auto-encoder learns its parameters by minimizing the reconstruction error. This minimization is usually done by stochastic gradient descent or alternatively regularization like “regularized auto-encoders” as in Bengio, Mesnil, Dauphin and Rifai (2013). Another unsupervised single layer learning algorithm which is used as a building block in constructing Deep Boltzmann Machine (DBM), which has many layers of hidden variables, and has no connections between the variables within the same layer (Salakhutdinov & Hinton, 2009); consequently this is a special case of the general Boltzmann machine (BM). In a DBM, each layer captures complicated, higher-order

correlations between the activities of hidden features in the layer below.

Hinton and Salakhutdinov (2011) present a generative deep learning model to acquire knowledge of the binary codes in documents. There are many layers in their deep learning network, where the lowest layer shows standard term frequency vector of the document and the highest layer shows the acquired binary code of the document. It is demonstrated that the binary codes of the documents that are semantically similar located relatively close in the Hamming space (Hinton & Salakhutdinov, 2011). According to the experimental results, using these binary codes in a deep learning architecture for document retrieval generates higher accuracy in a faster way compared to the standard term-frequency vector representation in semantic-based analysis.

Ranzato and Szummer (2008) present a semi-supervised deep learning model. In this study, they offer an algorithm to learn text document representations based on semi-supervised auto-encoders that are stacked to form a deep network. The model can be trained efficiently on partially labeled corpora, producing very compact representations of documents, while retaining as much class information and joint word statistics as possible. The authors show that for learning compact representations, deep learning models are better than shallow learning models because they need less computations and storage capacity.

Interacting with computers in natural language requires a representation of words, their meaning, and their meaning in context with other words. A recent development towards that goal are Vector Space Models (VSM) for words. VSMs represent each word as a high-dimensional vector, which is learned automatically from a large unannotated natural language corpus.

It is known that the distributed word representations in vector space group the similar words so that the using these representations work better on various problems in NLP (Mikolov, Sutskever, Chen, Corrado, and Dean, 2013b). In particular, Mikolov et al. develop *s* set of methods which is called word2vec for learning distributed vector representations of words in other words word embeddings. Their model is inspired from deep learning architectures and concepts but actually shallow neural networks are used for performance reasons (Mikolov, Chen, Corrado, & Dean, 2013a, 2013b). These word embeddings methods have recently attracted considerable attention in the academic platforms in terms of their ability to reveal interesting semantic relationships, specifically the analogy relations and their extreme scalability in terms of processing training data (Goldberg & Levy, 2014). Word embeddings vectors are actually based on the distributional hypothesis (Harris, 1954) which says, *"You shall know a word by the company it keeps"*. These models, specifically word2vec is a highly popular example of word embeddings and has received important attention in the scientific community within a short period with several extensions and improvements being published. The tool receives a text corpus as input and after the training, represents each word in the text as a relatively small sized dense vector. Size of the output vectors, usually indicated by *k*, is given as an input parameter and has a profound affect in the performance of systems using these vectors. Word2vec clusters semantically similar words in close coordinates in this *k* dimensional space. In order to find the coordinates of the words two neural network architectures are used, namely Continuous Bag of Words (CBOW) and skip-gram (SG). In the CBOW architecture, a word is predicted based on its context (left and right neighboring words in a fixed-length window, size of the windows is also an important input parameter). SG architecture works exactly in the opposite way; it tries to predict the context (right and left surrounding words of a specific word) by just looking at that specific word. In general, the SG architecture produces better word vectors for infrequent words while CBOW produces better results on a larger corpus. CBOW and SG uses two learning algorithms, namely hierarchical softmax (HS) and negative sampling (NS). The architectures of CBOW and SG are shown in Fig. 3. Generally, HS training algorithm produces good results for infrequent words while NS produces better results for frequent words. One of the greatest facilities of word2vec is that it makes possible arithmetic operations between word vectors. Embeddings have been used to improve NLP tasks such as NER, Part Of Speech (POS) Tagging, Dependency Parsing, and text classification.

Word embedding algorithms are able to extract semantic relations from very large amounts of textual documents generally with the help of feed-forward artificial neural networks. However, they need very large amount of textual data to perform reasonably. Studies have also been carried out to improve the semantic relations between the words that these models reveal by using external semantic sources. In one of these studies, Wikipedia is used as an external semantic source, aiming to better work on word2vec-like distributed approaches to problems involving small amount of labeled data (Wang et al., 2015). In another study (Cao, Li, Liu, Li, & Ji, 2015), an *n*-gram based and word-based artificial neural network is presented to model topics similar to the general topic modeling method with Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003). This unsupervised method has achieved high accuracy in the text mining algorithms used for topic identification/extraction. In another study, labels of semantic features, which were produced by human experts, were used to solve the problem of requiring very large training set requirement of word embedding algorithms such as word2vec and to work well in small data sets (Hill and Korhonen, 2014).

Word2vec and similar distributed word embedding methods are expected to improve the performance of standard text classification algorithms since they provide richer and more intense representation of documents than classical vector space representation. In this case, however, there are various problems with the use of the labeled textual data during the creation of distributed word representations. The most important of these is that distributed word-based learning algorithms do not have a intrinsic mechanism to use class labels in the training set used for training in classification algorithms. In this respect, these algorithms can be considered as unsupervised algorithms. To provide a partial solution to this problem, a classification training set can be used for training of distributed word-based learning algorithms. Thus, semantic relations between words in documents separated into different classes can be expected to indirectly reflect class labels to some degree. However, since the training sets used for classification in real life problems are usually very limited and the distributed word-for-word methods require very large data sets for training (Mikolov et al., 2013a); the power to reveal semantic relationships will weaken.

In one of the first studies to use word embeddings, a new technique called paragraph vectors is proposed instead of Bag of Words. Paragraph vectors is an unsupervised framework (Le & Mikolov, 2014) and continuously learns distributed word representations from textual materials. In this model, the paragraph vectors are formed by the combination of many word vectors in that paragraph and predict the next word in context. In this study, the authors offer two-paragraph vector algorithms, namely distributed memory model

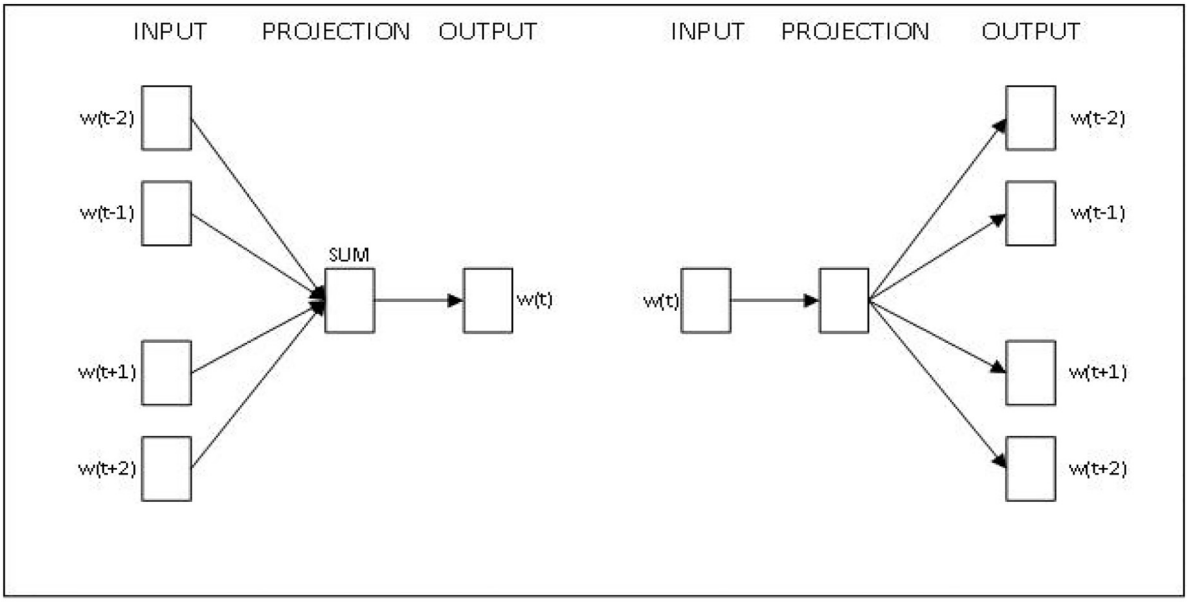


Fig. 3. (a) The CBOW architecture predicts a specific word based on its surrounding context and (b) the SG architecture predicts the surrounding words given the current word. (Adapted from Mikolov et al., 2013b).

and distributed bag of words (Fig. 4). A standard framework for learning word vectors is shown in Fig. 4(a). Context of three words ("the," "cat," and "sat") is used to predict the fourth word ("on") by calculating the average of these input vectors.

A framework for learning paragraph vector, namely distributed memory model: This framework is similar to the framework presented in (a); the only change is the additional paragraph token that is mapped to a vector via matrix *D*. In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word.

Distributed Bag of Words version of paragraph vectors: In this version, the paragraph vector is trained to predict the words in a small window without word ordering. Paragraph vectors can be fed directly to a machine learning algorithm such as Support Vector Machines (SVM) or *k*-means. Paragraph vectors were used in the field of sentiment analysis, error rate has been improved by more than 16% compared to complicated text classification methods, and an improvement about 30% compared to the simple BOW approach is achieved. Le and Mikolov (2014) perform a series of experiments on sentiment analysis and information retrieval tasks to show the effective behavior of the paragraph vectors. Their experimental setup and results are shown in Table 3. According to the experimental results reported in Le and Mikolov (2014), paragraph vector technique is superior to the conventional state-of-the-art BOW based text classification algorithms such as NB and SVM.

In a recent study, word2vec was used as a preliminary method with the hypothesis that it could improve the performance of classification by extracting semantic relations before text classification (Lilleberg, Zhu, & Zhang, 2015). This is compared with the tf-idf word weighting method, which is a widely used unsupervised transformation method. In this study, the skip-gram approach in the word2vec package was trained with a default vector length of 100, and the sum of weighted word vectors according to their frequency in the document is used for the representation of documents. In addition, the filtering and the usage of cutting words and weighting word2vec representations with tf-idf weighting technique have been tried. Lilleberg et al. (2015) use scikit-learn[5] Python library to load 20Newsgroups posts as a list of raw texts. Their experimental setup and results are shown in Table 3. According to their experimental results, they conclude that the combination of Word2vec weighted by tf-idf without stop words and tf-idf without stop words can result in better scores.

In a similar study (Turian, Ratinov, & Bengio, 2010); unsupervised vector space representations were used as additional features to enhance the accuracy of a supervised NLP system for Named Entity Recognition (NER). It has been reported that three different types of unsupervised word embeddings techniques and their combinations increase the accuracy of classification systems. In this study, high-performance word embeddings methods such as word2vec are not used. A similar approach is used in Taşpınar, Ganiz, and Acarman (2017), word embeddings are used in a novel way to extract features and augment train machine learning based classifiers to classify words to named entities in Named Entity Recognition (NER) setting. Inspired by these features, a similar study is conducted to create text classifiers using document embedding based features and some classifiers that operate on document embedding space in Çelenli, Öztürk, Şahin, Gerek, and Ganiz (2018).

In another semi-supervised NLP application, word embeddings were used to improve the performance of WSD algorithms (Taghipour & Ng, 2015). This study is generally classified as a semi-supervised framework since supervised NER algorithm was used

[5] http://scikitlearn.org/stable/datasets/twenty/newsgroups.html.

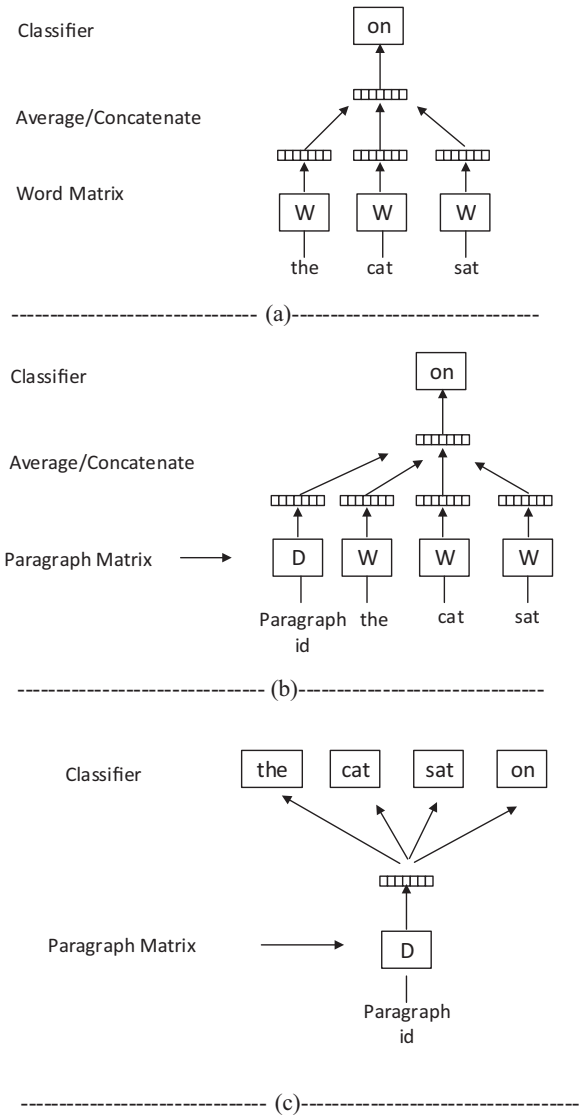


Fig. 4. (a) A framework for learning word vectors (b) A framework for distributed memory model (c) A framework for distributed bag of words (Adapted from Le & Mikolov, 2014).

with unsupervised word embeddings algorithms. The developed system in this study is the first to integrate word embeddings into WSD with two different methods; and it is evaluated in domain specific lexical tasks with SensEval / SemEval lexical and all-words tasks (Collobert & Weston, 2008). In this study, it was stated that word classification should include specific discriminative information for the classification problem; and for this purpose, an artificial neural network using lookup tables that contain word-case (POS) matches specifically for the WSD problem has replaced word vectors. In addition, since there are few training examples for each word in the WSD problem, the hidden layer is disabled to reduce model size and overfitting, as opposed to the original word segmentation model in the artificial neural network. The output layer of the artificial neural network calculates the conditional probability on labels when input text is given using class labels (the perception tag in the WSD field) using a SoftMax formula.

In the training of the artificial neural network, Stochastic Gradient Descent (SGD) and error return distribution are used to minimize the negative log-likelihood cost function. Because the main purpose in the study is to adapt the word embeddings into the problem using the artificial neural network, the reference table layer parameters are assigned with pre-trained word embeddings. Open source IMS tool for WSD is used. This software generates three types of features specific to WSD and uses the SVM algorithm as a classification algorithm. The authors prefer the above-mentioned study (Turian et al., 2010) to use the word embeddings learned by the artificial neural network in the classification system in their framework. Experimental results have shown that the developed system generates better results than the baseline systems.

Tang et al. (2014, 2016) also conduct experiments on emotion analysis related to the use of word embeddings in the field of classification. Two artificial neural networks were trained together; first, one to learn word embeddings and the next one is to predict

Table 3
Performance comparison of deep learning approaches across different datasets.[a]

Author	Year	Approach	Dataset	Performance metric	Results
Ranzato and Szummer	2008	An approach based on semi-supervised auto-encoders that are stacked to form a deep network	20 Newsgroups: 11,314 training 7531 test	Average accuracy (50%training)	20NewsGroups dataset:66
Hinton and Salakhutdinov	2011	a generative deep learning model to acquire knowledge of the binary codes in documents (Hybrid 128 − bit DGM using TF − IDF)	20NewsGroups (20 different newsgroups): 8314 training 3000 validation 7531 test articles The Reuters Corpus Volume II: 302,207 training 100,000 validation 402,207 test	Average accuracy (44%training)	20NewsGroups dataset:78 Reuters dataset:50
Mikolov et al.	2013a	Skip-gram model Continuous BOW model	Google News dataset (with 6 billion tokens)	Average accuracy	CBOW:63.7 Skip-gram:65.6
Mikolov et al.	2013b	several extensions that improve both the quality of the vectors and the training speed of continuous Skip-gram model	an internal Google dataset with one billion words	Average accuracy	72
Le & Mikolov	2014	an unsupervised learning algorithm that learns vector representations for variable length pieces of texts such as sentences and documents	Stanford sentiment treebank dataset[b] and IMDB dataset	Average accuracy	87.8
Tang et al.	2014	NRC-ngram and Combinations of different word embedding techniques	Benchmark Twitter SemEval 2013 sentiment classification dataset	Macro-F1 score	NRC:84.73 Uni + bi + tri word embedding:84.98
Cao et al.	2015	a novel neural topic model (NTM) where the representation of words and documents are efficiently and naturally combined into a uniform framework	20 Newsgroups(Train: 11,149, Test: 7403) Wiki10 + (Training: 11,550, Test: 5775) (Zubiaga, 2012) and Movie review data (Training: 3337,Test: 1669) (Pang & Lee, 2005)	Average accuracy	75.2
Wang et al.	2015	novel method of jointly embedding knowledge graphs and a text corpus so that entities and words/phrases are represented in the same vector space	19,544 word analogies[c]; 3,218 phrase analogies[d]	Average accuracy	Triplet classification:90 Phrases Analogical Reasoning Task:65 Words Analogical Reasoning Task:89.9
Lilleberg et al.	2015	combination of Word2vec weighted by tf-idf without stop words and tf-idf without stop words	20Newsgroups	Average accuracy	89.7

[a] The results may vary, even if the same data set is used because different preprocessing methods are used.
[b] http://nlp.Stanford.edu/sentiment/.
[c] code.google.com/p/word2vec/source/browse/trunk/questions-words.txt.
[d] code.google.com/p/word2vec/source/browse/trunk/questions-phrases.txt.

the emotion class. One of the main problems here is that very large data sets are required to learn word embeddings, but the training sets use for emotion classification are not in these sizes. To solve this problem, the authors have tagged five million positive and five million negative tweets using expressive emotional words (emotions) such as :-) and :-(to extend the tagged emotional data set. Although tagging Tweets in this way is not very successful and the training set is largely corrupted with noise, the authors have shown that this dataset and approach learn word embeddings that increase performance somewhat compared to the best features in Tang et al. (2016). In this study, artificial neural networks were applied which include the basic nerve layers of lookup → linear → hTanh → linear → softmax. Tang et al. (2016) mention that they empirically verify the effectiveness of sentiment embeddings on three sentiment analysis tasks: (1) on word level sentiment analysis, they show that sentiment embeddings are useful for discovering similarities between sentiment words. (2) On sentence level sentiment classification, sentiment embeddings are helpful in capturing discriminative features for predicting the sentiment of sentences. (3) On lexical level task like building sentiment lexicon, sentiment embeddings are shown to be useful for measuring the similarities between words. They conclude that hybrid models that capture both context and sentiment information are the best performers on all three tasks (Tang et al., 2016).

The semi-supervised studies based on word embeddings in the literature (Turian et al., 2010; Severyn & Moschitti, 2015a; Taghipour & Ng, 2015) have proposed approaches based on the use of word embeddings learned from a large unlabeled corpus as features in supervised classification systems.

4.3. Performance comparison of deep learning approaches

Table 3 presents the deep-learning approaches including their methodologies, experiment settings and experimental results.

It is possible to compare works that have used the same dataset. For example, Hinton and Salakhutdinov (2011) and Ranzato and Szummer (2008) use the same 20 Newsgroups dataset and both of them use accuracy as the performance evaluation metric. According to the experimental results reported in Table 3, generative deep learning model to acquire knowledge of documents' binary codes in Hinton and Salakhutdinov (2011) reaches higher classification accuracy with less training samples in compare to the study in Ranzato and Szummer (2008) which uses an approach based on semi-supervised auto-encoders that are stacked to form a deep network. Furthermore, Cao, Li, Liu, Li, and Ji (2015) seem to get higher classification accuracy than Ranzato and Szummer (2008) on 20 Newsgroups dataset with similar training and test sizes. This may be explained by Cao et al.'s novel Neural Topic Model (NTM) where the representation of words and documents are efficiently and naturally combined into a uniform framework. It should also be noticed that, Mikolov et al.'s Continuous BOW model is superior to Mikolov et al.'s Skip-gram model on Google News dataset. It is also very important to observe that several extensions in Mikolov et al. (2013b) improve both the quality of the vectors and the training speed of continuous Skip-gram model. Furthermore, according to the experimental results reported in Table 3, combination of Word2vec weighted by tf-idf without stop words advance the classification performance on 20 Newsgroups dataset. It is possible to compare deep-learning based approaches with corpus-based approaches that have used the same dataset. For instance Ganiz et al. (2009, 2011) reach an impressive classification accuracy on 20 Newsgroups dataset although it uses very small training size (i.e., 5% of the whole corpus) in comparision to deep-learning based studies (Cao et al., 2015; Hinton & Salakhutdinov, 2011; Lilleberg et al., 2015; Ranzato & Szummer, 2008) which uses the same dataset with more than 40% training size.

5. Word/character sequence enhanced systems

5.1. Overview of the approach

Words are treated as string sequences in these kinds of textual data representations. The main logic behind the algorithms in this category depends on a word/character sequence taken out from documents by ordinary string-matching method. *N*-gram based demonstration (Cavnar & Trenkle, 1994) and similar works in Ho and Funakoshi (1998), Ho and Nguyen (2000) and Fung (2003) are traditional examples of these types of systems.

5.2. Text classification studies of word/character sequence enhanced systems

This section provides an overview of word/character sequence enhanced systems offered for text classification; including their descriptions, experimental results and advantages/disadvantages.

According to Zipf's (1949) law *"The n^{th} most common word in a human language text occurs with a frequency inversely proportional to n"*. In other words, given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table. This law is explicitly true for general words that are common in all documents without any capability of differentiating between documents and this law is also true for specific words that are special for certain domains or subjects of documents. According to the discussions in Cavnar and Trenkle (1994), Zipf's law also implies that documents from the same category are observed to have similar *n*-gram frequency distributions. Furthermore, one of the most important difficulties in text classification is the existence of different types of textual errors, such as spelling and grammatical errors in emails, documents or forum messages. An accurate text classifier must handle these kinds of situations and must work reliably on all input. After analyzing Zipf's law, Cavnar and Trenkle (1994) present an *n*-gram based methodology for text classification, which has the capability to cope with textual errors. The main tasks in this methodology are generating *n*-gram frequency profiles, comparing/ranking *n*-gram frequency profiles, measuring distances between profiles and classification. Measuring distance between profiles of documents is a very simple process: firstly the *n*-grams for each category and for the test document are ranked from the most frequent to the least frequent and then the

distance for each *n*-gram to the corresponding *n*-gram in a selected category is measured, then all these distances are summed and the test documents is classified with the category which has the minimum distance to it. An illustrative example is given in Fig. 5.

This algorithm uses examples of the desired categories rather than using more complex and costly approaches like natural language parsing or assembling detailed lexicons. Principally this method defines a "categorization by example" process. Collecting examples and building profiles can even be controlled in a largely automatic way. Besides, this system is claimed to be robust, since it is based on the statistical properties of *N*-gram occurrences and not on any particular occurrence of a word. For example, it achieves 99.8% classification accuracy in one test case on Usenet newsgroup articles written in different languages according to the experimental results reported in Cavnar and Trenkle (1994). The authors also mention some future directions to improve the system's classification performance in cases where it does not work well.

Peng and Schuurmans (2003) present a chain augmented NB classifier (CAN). It depends on statistical *n*-gram language modeling and gets dependence between adjacent attributes as a Markov chain. They obtain further performance improvements through better smoothing techniques than Laplace smoothing. Their CAN Bayes modeling method has the capability to work at either character level or word level, which offers language independent abilities to cope with Eastern languages like Japanese and Chinese. There are two advantages of CAN over standard NB classifiers: (1) A chain augmented NB model relaxes some of the independence assumptions of NB. (2) Smoothing methods from statistical language modeling can be used to recover better estimates than the Laplace smoothing methods usually used in NB classification.

Their empirical evaluation includes Greek authorship attribution dataset with 10 Greek authors, 20NewsGroups dataset with 20 categories and Chinese TREC topic detection dataset with six topics. They get extensive improvements over standard NB classification on these three real world data sets as reported in Table 4. In another study, Suzuki, Tsai, Ishida, Goto, and Hirasawa (2009) refine the database of feature words with the help of mutual information and frequency ratio in documents. They show the efficiency of the suggested technique by several experiments on Reuters-21578 dataset. They analyze the difference between Word *N*-gram and Character *N*-gram in the case of Chinese and Japanese and state that Word *N*-gram is more effective than Character *N*-gram. Furthermore, their experiments make clear that the results of Chinese are 10% or lower compared to English and Japanese, even if they use Word *N*-gram. Still, there are cases that they cannot explain in detail, which is left for future work.

In Rostami and Mumivand (2014), a novel algorithm has been offered for automatic text classification. There are two steps, namely text pre-processing and text classification. Text preparation, indexing and indices weighting, stemming, filtering stop words are performed in pre-processing step. Additionally, they use *N*-gram technique for indexing and tf-idf technique for weighting the indices in the pre-processing step. In the text classification step, *k-NN* is applied in order to train the model for classifying. They use precision and recall measures (i.e., Micro-F1 and Macro-F1) as their evaluation metrics. They perform experiments on Reuters-21578 data set. The experimental results show that their suggested method is superior to NB and DT algorithms on this data set.

A system is developed by Razon and Barnden (2015) which is actually a learner-focused text readability-indexing tool for second language learners of English. Student essays are used to regulate the system, making it capable of providing an accurate approximation of second language learners' real reading capacity spectrum. In this study, they present a comparative review of two semantic algorithms, namely, LSI and Concept Indexing (CI) for text content analysis. They show that incorporating POS *n*-gram features to approximate syntactic complexity of the text documents (CI) can improve the traditional LSI. Without the integration of POS *n*-gram features, the difference between their mean exact agreement accuracies (MEAA) can achieve as high as 23%, in favor of CI.

5.3. Performance comparison of word/character sequence enhanced systems

Table 4 presents the word/character sequence enhanced approaches including their methodologies, experiment settings and experimental results.

For word/character sequence enhanced approaches, it is practically challenging to use the same experimental environment. However, it could be easier to compare works that have used the same dataset. For example, Suzuki et al. (2009) and Rostami and Mumivand (2014) use the same Reuters-21578 dataset and both of them use Micro-F1 as the performance evaluation metric. The methodology offered in Suzuki et al. (2009), which is based on refinement of feature terms using character *n*-gram seems to be slightly superior to the methodology offered in Rostami and Mumivand (2014) where *k-NN* and *N*-Gram indexing techniques are applied. It is also worthy to notice that one of the first studies in word/character sequence enhanced approaches type get significant classification accuracy on USENET dataset.

6. Linguistic enriched methods

6.1. Overview of the approach

These approaches utilize syntactic and lexical rules to get the noun phrases, terminologies and entities from documents and enhance the representation using these linguistic units. For example, Papka and Allan (1998) take advantage of multi-words to increase the efficiency of text retrieval systems. Furthermore, Lewis (1992) makes a detailed analysis, which compares phrase-base indexing and word-based indexing for representation of documents.

6.2. Text classification studies of linguistic enriched approaches

Lewis (1992) studies the properties of phrasal and clustered indexing languages on text classification. According to this work,

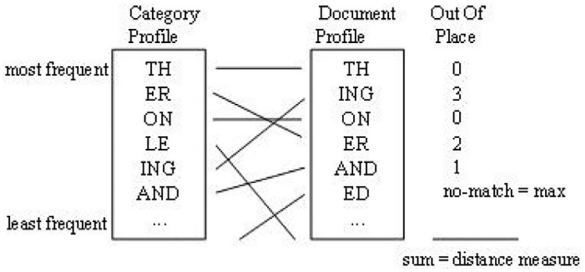


Fig. 5. An illustrative example for the calculation of *n*-gram profile distance between documents (Adapted from Cavnar & Trenkle, 1994).

optimal effectiveness occurs when using just a small percentage of the indexing words available and that effectiveness peaks at a higher feature set size, and there is lower effectiveness for syntactic phrase indexing than for word-based indexing. Lewis (1992) reports results proposing that a conventional word clustering approach is unlikely to offer considerably enhanced text demonstrations.

In another study (Papka & Allan, 1998), the effect of using multiword query features on text retrieval systems is investigated. The multiword features are modeled as a set of words appearing within windows of varying sizes. Their experiments indicate that better precision results are achieved when queries are expanded with features that are modeled as a set of words, which appear further, separated within natural language text.

There are also studies in this category for the sentiment classification problem. As Nasukawa and Yi (2003) state, there are three main tasks in sentiment classification: finding sentiment expressions in the available corpus, differentiating the polarity and the strength of these expressions and finding the correlations between these expressions and the subject. Unfortunately, successfully accomplishing these tasks is not an easy job because sentences using the same words can express different sentiments (Fei et al., 2004). In some of previous works phrase patterns are used for sentiment classification (Nasukawa & Yi, 2003; Pang, Lee, & Vaithyanathan, 2002; Turney, 2002). Fei et al. (2004) build a two-part methodology in their study.

Their methodology is composed of three steps:

(1) The first step is selecting words, which can express sentiment in the text. Secondly, they add tags to these selected words reflecting parts of speech and sentiment (positive or negative).
(2) The second step involves the construction of phrase patterns. These phrase patterns are composed of adjectives, nouns, adverbs...etc. They use 40 different phrase patterns.
(3) In the third step, they use a machine-learning algorithm to evaluate sentiment orientation. By this algorithm, the main goal is to identify the sentiment category (i.e., positive or negative) and the strength value (a value between -1 and 1) of a phrase pattern.

The sentiment classification step is the final step in their approach. In this step, they evaluate their methodology on 320 sports reviews that are collected from yahoo.com. Their experimental setup and results are reported in Table 5. They conclude that they will improve the way they construct phrase patterns as their future work.

In a more recent study, Abbasi, France, Zhang, and Chen (2011) mention some research gaps in existing studies in the sentiment classification domain. These gaps include using a limited set of n-gran features under at most two categories, computational difficulties of feature sets, performance degradation come with noisy feature sets, redundancy, missing incorporation of semantic

Table 4
Performance comparison of word/character sequence enhanced approaches across same/different datasets.[a]

Author	Year	Approach	Dataset	Performance metric	Results
Cavnar and Trenkle	1994	*N*-Gram	USENET(including 3713 language examples from the soc.culture newsgroup hierarchy of the Usenet)	Average classification accuracy	99.8
Peng and Schuurmans	2003	Combining *N*-Gram Language Models and NB	20 Newsgroup data (80% of the documents is used for training and the rest is used for testing)	Average classification accuracy	89.08
Suzuki et al.	2009	Refinement of Feature Terms and Increase in Classification Accuracy on Multilingual Text Categorization Using Character *N*-Gram	Reuters-21578	Macro -F1	92.3
Rostami and Mumivand	2014	k-NN and *N*-Gram Indexing	Reuters 21578	Micro-F1	91.46
Razon and Barnden	2015	Concept Indexing with Integrated POSN-Gram Features	English essays written by high school students (2010-2014) 2/3 of the essays is for training and the rest is for testing	Average classification accuracy	95.1

[a] The results may vary, even if the same data set is used because different preprocessing methods are used.

information. With the motivation to address these problems and fulfill these gaps, Abbasi et al. (2011) propose a Feature Relation Network (FRN), which uses rich set of *n*-gram features. A rule-based multivariate *n*-gram feature selection method powerfully eliminates redundant or less useful *n*-grams. This methodology also incorporates semantic information derived from existing lexical resources, enabling augmented weighting/ranking of *n*-gram features. Abbasi et al. (2011) compare the suggested FRN feature selection method against some of the existing works in the literature. According to experimental results reported in Abbasi et al. (2011) the suggested approach can improve opinion classification performance over existing selection methods.

6.3. Performance comparison of linguistic enriched approaches

Table 5 presents the linguistic enriched approaches including their methodologies, experiment settings and experimental results.

For linguistic enriched approaches listed in Table 5, it is not easy to compare the studies since they perform their experiments on different datasets; still maybe it is meaningful to compare the studies in Fei et al. (2004) and Abbasi et al. (2011) since they use review datasets. The algorithm suggested in Abbasi et al. (2011), uses rich set of *n*-gram features, seems to be accurate than the algorithm suggested in Fei et al. (2004) that uses phrase patterns.

7. Comparison of semantic text classification approaches

This survey investigates the existing and recent advancements in the semantic text classification field and highlights strengths in comparison to the traditional text classification approach. This section presents a summary comparison with respect to a number of key criteria.

Data Representation: In traditional text classification, data is represented with bag-of-words or bag-of-n grams because of their efficiency and simplicity. Moreover, in semantic text classification data representation is enriched by semantic graphs, statistical calculation derived from the corpus, outer knowledge sources such as WordNet, Wikipedia etc., deep learning approaches, word/character sequence enhanced approaches, linguistic enriched approaches.

Ontology or Background Source: In traditional text classification, no ontology or background source is used while an ontology or thesaurus is used by domain knowledge-based systems to identify concepts in documents. Examples of knowledge bases are dictionaries, thesauri and encyclopedic resources. Common knowledge bases are WordNet, Wiktionary and Wikipedia. They improve the demonstration of words by the utilization of semantic similarities among words.

Semantics: Traditional text classification algorithms concentrate on simple vector representation of words or phrases in documents which produces poor classification results because it ignores the hidden semantic connections between words and documents (Salton & Yang, 1973; Turney & Pantel, 2010). Conversely, semantic-based text classification focuses on the extraction of the semantic relationships between the terms and the documents, which consequently results more accurate classification performance.

Synonymy, polysemy: Traditional classification algorithms do not have the capability to capture synonymous, polysemyous words; since these algorithms only concentrate on the syntax of the textual materials. The semantic text classification algorithms resolve the problems of synonymy and polysemy by using semantic techniques like LSI, higher-order paths or class based term weights.

Ambiguity: Traditional classification algorithms do not have the capability to capture ambiguity; since these algorithms only concentrate on the syntax of the textual materials. On the other hand, the problem of ambiguity is resolved by the WSD techniques in semantic text classification.

Description: Traditional text classification methods have poor capabilities in explaining to its users why certain results are achieved. Because they cannot relate semantically nearby terms and they cannot explain how the result clusters are related to one another. In contrast, semantic text classification algorithms have the capability to locate the instances semantically and to explain and analyze the classification results.

Applicability in Social Networks: In social networks, textual data are huge, dynamic, and noisy and there are emoticons (smile, sad). So to achieve an accurate classification result, more than syntax-oriented algorithms are needed. Conversely, semantic text classification approaches are more suitable than traditional ones for usage in social networks because of their capability to get hidden connections between terms and documents.

Classification Accuracy: The classification accuracy of the traditional classification algorithms are poor due to their deficiencies related to synonymy, polysemy, ambiguity and semantics. On the other hand, it is proved and reported that semantic text classification algorithms achieve better classification accuracies than the traditional ones.

This survey covers five broad approaches to semantic text classification. Of these, knowledge-based, corpus-based approaches and deep-learning based approaches are the most widely used. Sections 2–6 present existing and recent techniques used in each of these approaches. The strengths and weaknesses of the most widely used approaches are summarized below.

7.1. Domain knowledge based approach

Strength:

- Knowledge-based approaches (enrich the representation of terms by utilizing semantic relatedness among terms with the help of ontology or thesaurus.
- There are domain specific knowledge bases, which are specialized in specific fields. For instance, Gene Ontology (GO) (The Gene Ontology Consortium, 2005) and the Medical Subject Headings[6] (MeSH) are biomedical vocabulary resources.

Table 5
Performance comparison of linguistic enriched approaches across different datasets.[a]

Author	Year	Approach	Dataset	Performance metric	Results
Lewis	1992	An Evaluation of Phrasal and Clustered Demonstrations on a Text Classification Field	Reuters (22,173 full-text newswire stories; 1987 stories are used for testing.)	Breakeven point Micro averaged recall/precision	65
Papka and Allan	1998	Document Classification Using Multiword Features	TREC-4	Breakdown of precision improvement in the test set	26.1
Fei et al.	2004	A sentiment classification system with phrase patterns	*yahoo.com (*170 of these reviews expresses positive sentiment while 150 of them expresses negative sentiment*)*	classification accuracy	for positive sentiment reviews:81.67% classification accuracy for negative sentiment reviews:91%
Abbasi et al.	2011	Feature Relation Network (FRN) which uses rich set of *n*-gram features	Epinions (www.epinions.com) Edmunds (www.edmunds.com)	classification accuracy	Epinions (digital cameras):88.42 Edmunds (Automobile):90.70

[a] The results may vary, even if the same data set is used because different preprocessing methods are used.

- A knowledge source such as WordNet groups nouns, adjectives, adverbs and verbs into synsets.

Weakness:

- Produces language dependent systems, so they cannot be applied to other languages.
- Only a few lexical databases such as WordNet and FameNet (dedicated to English), GermaNet (the German equivalent to WordNet) have currently been established. In general, most of the lexical databases are devoted to English, German and Chinese.
- There is no natural language processor that creates grammatical tags such as POS tags based on syntactic analysis, which can be used for documents in most languages.
- Such resources are expensive to maintain, and often unavailable in specific domains.
- The scope and the coverage of words in knowledge bases can limit the capability of methods. For instance, Wiktionary covers twice the amount of words than in WordNet.

7.2. Corpus-based approach

Strength:

- Corpus based systems are independent from any knowledge source such as WordNet, Wikipedia and may be compatible with any language.
- They do not require the processing of large external knowledge sources.
- Since corpus-based systems are generated from corpus-based statistics, they are always up-to-date.
- They do not have any coverage problem since the semantic relations between terms are specific to the domain of the corpus. Unstructured corpora are generally much easier to find and adapt.

Weakness:

- Processing a huge corpus of documents will be a very important problem to be considered with corpus-based systems, as it is a substantial computational cost.
- These methods individually address the syntactic and semantic features of documents, which may negatively affect the results.
- They do not measure lexical semantic relatedness since that requires a certain knowledge base about the terms.

7.3. Deep learning

Strength:

- Deep learning algorithms are advantageous especially for unsupervised or semi-supervised learning where there is large amount of unlabeled data, and typically learn data representations in a greedy layer-wise fashion.
- Has best-in-class performance on problems in text classification domain.
- Has an architecture that can be adapted to new problems relatively easily.

Weakness:

- Requires and performs better with a large amount of data and is unlikely to outperform other approaches with small amount of data.
- Is computationally expensive to train which requires extensive hardware resources.
- While other machine learning classification algorithms (e.g., decision trees, logistic regression etc.) can be understood, it is difficult to explain how DL models function.
- Deep learning models are extremely complex models with strong theoretical foundation which is hard to build and implement.

It is possible to make a global comparison for each type of algorithms on 20NewsGroups dataset and Reuters dataset. Across all the studies summarized in this paper, studies in Altınel et al. (2014a, 2015b), Cao et al. (2015), Ganiz et al. (2009, 2011), Hinton and Salakhutdinov (2011), Nasir et al. (2011), Peng and Schuurmans (2003), Poyraz et al. (2012, 2014), Ranzato and Szummer (2008), Siolas and d'Alché-Buc (2000) and Suganya and Gomathi (2013) use the 20NewsGroups dataset. Results of these are presented in Tables 1–5. Comparing these studies suggests that deep-learning and knowledge-based approaches are more effective. Deep Learning algorithms are reasonably advantageous especially for unsupervised or semi-supervised learning where there is huge amount of unlabeled data, and typically learn data representations in a greedy layer-wise fashion (Bengio et al., 2007; Hinton et al., 2006). Empirical studies have demonstrated that deep learning often generate higher classification accuracy than traditional machine learning algorithms in many domains like speech recognition, computer vision, NLP, bioinformatics (Larochelle et al., 2009). On the

[6] http://www.nlm.nih.gov/mesh/.

other hand, knowledge-based approaches (Nasir et al., 2011; Siolas & d'Alché-Buc, 2000; Suganya & Gomathi, 2013) take advantage of knowledge-based resources to enrich the text representation. Moreover; corpus-based studies, called language-independent systems since they are independent from any knowledge source such as WordNet and Wikipedia, also produce noticeable classification accuracies on the 20 newsgroups text dataset such as Altınel et al. (2014a, 2015b), Ganiz et al. (2009, 2011) and Poyraz et al. (2012, 2014) as shown in Table 2.

Similarly, studies in Rodriguez et al. (2000), Bloehdorn et al. (2006), Hinton and Salakhutdinov (2011), Lewis (1992), Liu et al. (2004), Rostami and Mumivand (2014), Suzuki et al. (2009) and Uysal and Gunal (2014) used the Reuters dataset, with results summarized in Tables 1–5. Comparing these studies suggests that corpus-based are more effective. For instance, Uysal and Gunal (2014) reached 88.4% classification accuracy with their GA oriented latent semantic features (GALSF). There are also studies in the knowledge-based approaches category get noticeable classification accuracies such as Liu et al. (2004) which offer Local Relevancy Weighted LSI algorithm. Besides, the number of presented word/character sequence enhanced approaches is relatively less in compare to the other categories, still according to the experimental results reported in Table 4, they generate very significant F1-scores such as Rostami and Mumivand (2014) and Suzuki et al. (2009).

8. Current challenges, future directions and concluding remarks

Implementing a semantic text classification algorithm has several challenges for researchers:

- *Availability of a knowledge base for a specific language:* Only a small number of lexical databases are available for a limited number of languages (i.e., English, German, and Chinese etc.). Many languages do not have their lexical databases. In other words, only for these specific languages knowledge-based systems can be generated. Moreover, knowledge-based systems for one specific language cannot be effectively used for another language, so they are mostly language-dependent and target general domain. Such resources are usually expensive to maintain due to the constantly evolving nature of the language and often unavailable in specific / technical domains. However, since they enhance the classification performance, researchers need to be encouraged to build knowledge bases for the remaining languages, or some automatic converters should be implemented.
- *Processing complexity of a large external knowledge base:* There exists a considerable processing cost for both knowledge-based systems and corpus-base systems. This processing cost includes the processing of a large external knowledge base for knowledge-based systems and pre-processing of the massive corpus for the corpus-based systems. Unfortunately, this extra cost increases as the size of the corpus increases. Researchers who implement corpus-based or knowledge-based systems need to optimize the processing time/complexity of their algorithms.
- *Complexity of computations to extract latent semantics:* Corpus-based systems use usually expensive mathematical calculations in order to extract knowledge i.e., latent semantics in the training corpus. These computations will increase the overall complexity of the classification system and hence the running time of the algorithm. To overcome this challenge, researchers need to come up ways and to reduce complexity of their algorithms.
- *Accessing massive amounts of unlabeled data:* Deep learning (DL) algorithms are advantageous especially for unsupervised or semi-supervised learning when there exists huge amounts of unlabeled data, and typically learn data representations in a greedy layer-wise fashion. It is not always possible to access or collect the amount of data to benefit the advantage of DL algorithms. Collecting, storing, and curating this amount of data is also associated with several different costs.
- *Computational of hardware systems:* Making a DL system work is computationally expensive. Especially, training a DL system requires expensive hardware resources. This should be taken into consideration when starting a DL based project.
- *Analysis:* For several machine learning based classification algorithms (e.g., decision trees, logistic regression etc.) it is possible to understand and explain the learned model and more importantly the decisions given by this model. This leads to several problems in the real world applications such as law enforcement and health. Furthermore, the researchers should be aware of the privacy related public discussions and regulations.

We advise researchers to apply different text mining techniques to pre-process and to filter the data in knowledge-based and corpus-based systems. In addition, different machine learning algorithms need to be utilized to filter unrelated information from the large text corpora. Nevertheless, determining whether to use a knowledge-based or corpus-based approach for semantic text classification is still a challenging task that depends on availability and the size of the dataset and knowledge bases, and the nature of the problem being investigated. In the future, text-mining tools can also be used as intelligent agents that can extract latent semantics from textual materials and report the relevant analysis results to the users without requiring an explicit request.

Many technologies have been developed for text classification using different machine learning algorithms. However, text classification is more challenging since in the text words have semantic connections, which are far from easy to model using computers. Extracting semantic connections from such unstructured form is a critical task for the success of text classification systems. We think that choosing right semantic text classification method depends on a number of inter-related factors. Each category of method has certain benefits over others but at the mean time suffers from certain restrictions, as described above.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2018.08.001.

References

Aas, K., & Eikvil, L. (1999). Text categorization: A survey.

Abbasi, A., France, S., Zhang, Z., & Chen, H. (2011). Selecting attributes for sentiment classification using feature relation networks. *IEEE Transactions on Knowledge and Data Engineering, 23*(3), 447–462.

Aggarwal, C. C., & Zhai, C. (2012). *A survey of text classification algorithms. Mining text data.* US: Springer163–222.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pascal, M., & Soroa, A. (2009). A study on similarity and relatedness using distributional and WordNet-based approaches. *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics (NAACL'09)* (pp. 19–27). Association for Computational Linguistics.

Alpaydın, E. (2004). *Introduction to machine learning.* MIT Press.

Altınel, B., Ganiz, M. C., & Diri, B. (2014a). A semantic kernel for text classification based on iterative higher–order relations between words and documents. *Proceedings of the thirteenth international conference on artificial intelligence and soft computing (ICAISC), Lecture Notes in Artificial Intelligence (LNAI), 8467* (pp. 505–517). .

Altınel, B., Ganiz, M. C., & Diri, B. (2014b). A simple semantic kernel approach for SVM using higher-order paths. *Proceedings of the IEEE international symposium on innovations in intelligent systems and applications (INISTA)* (pp. 431–435). .

Altınel, B., Ganiz, M. C., & Diri, B. (2015a). A corpus-based semantic kernel for text classification by using meaning values of terms. *Engineering Applications of Artificial Intelligence, 43*(1), 54–66.

Altınel, B., Diri, B., & Ganiz, M. C. (2015b). A novel semantic smoothing kernel for text classification with class-based weighting. *Knowledge-Based Systems, 89*(1), 54–66.

Bai, X., Padman, R., & Airoldi, E. (2004). *Sentiment extraction from unstructured text using tabu search-enhanced Markov blanket.* Carnegie Mellon University, School of Computer Science [Institute for Software Research International].

Balinsky, A., Balinsky, H., & Simske, S. (2011a). On the Helmholtz principle for data mining. *Proceedings of conference on knowledge discovery*.

Balinsky, A., Balinsky, H., & Simske, S. (2011b). Rapid change detection and text mining. *Proceedings of second conference on mathematics in defense (IMA)*Defense Academy.

Bengio, Y., Mesnil, G., Dauphin, Y., & Rifai, S. (2013). Better mixing via deep representations. *Proceedings of the 30th international conference on machine learning (ICML-13)* (pp. 552–560). .

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems, 19*, 153.

Berry, M. W. (2004). Survey of text mining. *Computing Reviews, 45*(9), 548.

Biricik, G., Diri, B., & Sönmez, A. C. (2009). A new method for attribute extraction with application on text classification. *Proceedings of the fifth international IEEE conference on soft computing, computing with words and perceptions in system analysis, decision and control (ICSCCW)* (pp. 1–4). .

Biricik, G., Diri, B., & Sönmez, A. C. (2012). Abstract feature extraction for text classification. *Turkish Journal of Electrical Engineering & Computer Sciences, 20*(1), 1137–1159.

Bisson, G., & Hussain, F. (2008). Chi-sim: a new similarity measure for the co-clustering task. *Proceedings of the seventh international conference on machine learning and applications* (pp. 211–217). .

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research, 3*, 993–1022.

Bloehdorn, S., Basili, R., Cammisa, M., & Moschitti, A. (2006). Semantic kernels for text classification based on topological measures of feature similarity". *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.* Springer307–318.

Campbell, C. (2002). Kernel methods: A survey of current techniques. *Neurocomputing, 48*(1), 63–84.

Cao, Z., Li, S., Liu, Y., Li, W., & Ji, H. (2015). A novel neural topic model and its supervised extension. *Proceedings of the AAAI* (pp. 2210–2216). .

Cavnar, W. B., & Trenkle, J. M. (1994). *N*-gram based text categorization. *Proceedings of third annual symposium on document analysis and information retrieval* (pp. 161–169). .

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the twenty-fifth international conference on machine learning* (pp. 160–167). ACM.

Çelenli, H.İ., Öztürk, S. T., Şahin, G., Gerek, A., & Ganiz, M. C. (2018). Document embedding based supervised methods for Turkish text classification. *Proceedings of the international conference in computer science and engineering (UBMK'18)*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing, 3*, e2.

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing, 7*(3–4), 197–387.

Dos Santos, C. N., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. *Proceedings of the COLING* (pp. 69–78). .

Edwards, A., & Pottenger, W. M. (2011). Higher-order Q-learning. *Proceedings of the IEEE symposium adaptive dynamic programming and reinforcement learning* (pp. 128–134). .

Elavarasi, S. A., Akilandeswari, J., & Menaga, K. (2014). A survey on semantic similarity measure. *International Journal of Research in Advent Technology, 2*(4), 389–398.

Fei, Z., Liu, J., & Wu, G. (2004). Sentiment classification using phrase patterns. *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on* (pp. 1147–1152). IEEE.

Fung, B. C. M. (2003). Hierarchical document clustering using frequent itemsets. *Proceedings of SIAM international conference on data mining* (pp. 59–70). .

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the IJCAI. Vol. 7. Proceedings of the IJCAI* (pp. 1606–1611).

Ganiz, M. C., Kanitkar, S., Chuah, M. C., & Pottenger, W. M. (2006). Detection of interdomain routing anomalies based on higher-order path analysis. *Proceedings of the sixth IEEE international conference on data mining* (pp. 874–879). .

Ganiz, M. C., Lytkin, N. I., & Pottenger, W. M. (2009). Leveraging higher- order dependencies between features for text classification. *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 Transactions on Knowledge and Data Engineering, 23*(7), 1022–1034.

Ganiz, M. C., Tutkan, M., & Akyokuş, S. (2015). A novel classifier based on meaning for text classification. *Proceedings of the innovations in intelligent systems and applications*.

Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision*Stanford University1–12 Technical report.

Goldberg, Y., & Levy, O. (2014). Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722.

Halavais, A., & Lackaff, D. (2008). An analysis of topical coverage of Wikipedia. *Journal of Computer-Mediated Communication, 13*(2), 429–440.

Harrington, B. (2010). A semantic network approach to measuring relatedness. *Proceedings of the twenty-third international conference on computational linguistics* (pp. 356–364). Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2–3), 146–162.

He, J., Ostendorf, M., He, X., Chen, J., Gao, J., Li, L., et al. (2016). Deep reinforcement learning with a combinatorial action space for predicting popular reddit threads. arXiv:1606.03667.

Hill, F., & Korhonen, A. (2014). Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean. *Proceedings of empirical methods in natural language processing (EMNLP)* (pp. 255–265). .

Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length and Helmholtz free energy. *Proceedings of the advances in neural information processing systems* (pp. 3–10). .

Hinton, G. E., Osindero, S., & The, Y-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computing, 18*(7), 1527–1554.
Hinton, G., & Salakhutdinov, R. (2011). Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science, 3*(1), 74–91.
Ho, T. B., & Funakoshi, K. (1998). Information retrieval using rough sets. *Journal of the Japanese Society for Artificial Intelligence, 13*(3), 424–433.
Ho, T. B., & Nguyen, N. B. (2000). Non-hierarchical document clustering based on a tolerance rough set model. *International Journal of Intelligent Systems, 17*(1), 199–212.
Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. *Proceedings of Ldv Forum. Vol. 20. Proceedings of Ldv Forum* (pp. 19–62).
Hu, S., Zuo, Y., Wang, L., & Liu, P. (2016). A review about building hidden layer methods of deep learning. *Journal of Advances in Information Technology, 7*(1), 13–22.
Hughes, T., & Ramage, D. (2007). Lexical semantic relatedness with random graph walks. *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* (pp. 581–589). Association for Computational Linguistics.
Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., et al. (2015). A survey on text mining in social networks. *The Knowledge Engineering Review, 30*(02), 157–170.
Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2007). A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology, 51*(6), 343–358.
Jarmasz, M., & Szpakowicz, S. (2003). Roget's thesaurus and semantic similarity. *Proceedings of conference on recent advances in natural language processing (RANLP 2003)* (pp. 212–219).
Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2004). Learning semantic similarity. *Advances in Neural Information Processing Systems, 15*(1), 657–664.
Kim, K., Chung, B. S., Choi, Y., Lee, S., Jung, J. Y., & Park, J. (2014). Language independent semantic kernels for short-text classification. *Expert Systems with Applications, 41*(2), 735–743.
Kontostathis, A., & Pottenger, W. M. (2006). A framework for understanding LSI performance. *Journal of Information Processing and Management, 12*(1), 56–73.
Kozima, H., & Furugori, T. (1993). Similarity between words computed by spreading activation on an English dictionary. *Proceedings of the sixth conference on European chapter of the association for computational linguistics (EACL '93)* (pp. 232–239). Association for Computational Linguistics.
Kunze, C., & Lemnitzer, L. (2002). GermaNet – representation, visualization, application. *Proceedings of the international conference on language resources and evaluation (LREC'02), Las Palmas, Spain* (pp. 1485–1491). ELRA.
Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research, 10*, 1–40.
Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International Conference on Machine Learning* (pp. 1188–1196). .
Leacock, C., Towell, G., & Voorhees, E. (1993). Corpus-based statistical sense resolution. *Proceedings of the ARPA workshop on human language technology* (pp. 260–265).
Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In C. FellBaum (Ed.). *WordNet: An electronic lexical database* (pp. 305–332). Cambridge, MA: MIT Press.
LeCun, Z., Bengio, Z., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.
Lee, M. D., Pincombe, B., & Welsh, M. (2005). An empirical evaluation of models of text document similarity. *Proceedings of the CogSci* (pp. 1254–1259). .
Lewis, D. D. (1992). An evaluation of phrasal and clustered representation on a text categorization task. *Proceedings of the fifteenth annual international ACM SIGIR conference on research and development in information retrieval* (pp. 37–50). .
Li, Y., Bandar, Z., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering, 15*(4), 871–882.
Li, S., Wu, T., & Pottenger, W. M. (2005). Distributed higher-order association rule mining using information extracted from textual data. *SIGKDD Explorations Newsletter Natural Language Processing and Text Mining, 7*(1), 26–35.
Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and Word2vec for text classification with semantic features. *Proceedings of the fourteenth IEEE international conference on cognitive informatics & cognitive computing (ICCI* CC)* (pp. 136–140). IEEE.
Liu, T., Chen, Z., Zhang, B., Ma, W. Y., & Wu, G. (2004). Improving text classification using local latent semantic indexing. *Proceedings of the fourth IEEE international conference on data mining (ICDM'04)* (pp. 162–169). .
Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Journal of Expert Systems with Applications, 38*(1), 12708–12716.
Lytkin, N. (2009). *Variance-based clustering methods, higher-order data transformations, and their applications*[Ph.D. Thesis]. NJ: Rutgers University.
Meyer, C., & Gurevych, I. (2010). How web communities analyze human language: Word senses in Wiktionary. *Proceedings of the second web science conference*.
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Proceedings of the advances in neural information processing systems* (pp. 3111–3119). .
Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). *Five papers on WordNet*Stanford University Technical report.
Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Journal of Computational Linguistics, 17*(1), 21–48.
Naik, M. P., Prajapati, H. B., & Dabhi, V. K. (2015). A survey on semantic document clustering. *Proceedings of the IEEE International conference on electrical, computer and communication technologies (ICECCT)* (pp. 1–10). IEEE.
Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data, 2*(1), 1.
Nasir, J. A., Karim, A., Tsatsaronis, G., & Varlamis, I. (2011). *A knowledge-based semantic kernel for text classification, string processing and information retrieval.* Springer261–266.
Nasir, J. A., Varlamis, I., Karim, A., & Tsatsaronis, G. (2013). Semantic smoothing for text clustering. *Knowledge-Based Systems, 54*(1), 216–229.
Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. *Proceedings of the second international conference on knowledge capture* (pp. 70–77). ACM.
Navarro, E., Sajous, F., Gaume, B., Pr´evot, L., ShuKai, H., Tzu-Yi, K., et al. (2009). Wiktionary and NLP: Improving synonymy networks. *Proceedings of the 2009 workshop on the people's web meets NLP: Collaboratively constructed semantic resources (People's Web '09)* (pp. 19–27). Association for Computational Linguistics.
Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence, 193*, 217–250.
Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine-learning techniques. *Proceedings of the ACL-02 conference on empirical methods in natural language processing. Vol. 10. Proceedings of the ACL-02 conference on empirical methods in natural language processing* (pp. 79–86). Association for Computational Linguistics.
Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the forty-third annual meeting on association for computational linguistics* (pp. 115–124). .
Pantel, P., Crestan, E., Borkovsky, A., Popescu, A., & Vyas, V. (2009). Web-scale distributional similarity and entity set expansion. *Proceedings of the 2009 conference on empirical methods in natural language processing (EMNLP '09)* (pp. 938–947). Association for Computational Linguistics.
Papka, R., & Allan, J. (1998). Document classification using multiword features. *Proceedings of the seventh international conference on information and knowledge management table of contents* (pp. 124–131).
Peng, F., & Schuurmans, D. (2003). Combining naive Bayes and n-gram language models for text classification. *Proceedings of the European conference on information retrieval* (pp. 335–350). Springer.
Poyraz, M., Kilimci, Z. H., & Ganiz, M. C. (2012). A Novel semantic smoothing method based on higher-order paths for text classification. *Proceedings of the IEEE international conference on data mining (ICDM)* (pp. 615–624). .
Poyraz, M., Kilimci, Z. H., & Ganiz, M. C. (2014). Higher-order smoothing: A novel semantic smoothing method for text classification. *Journal of Computer Science and Technology, 29*(3), 376–391.
Ranzato, M., & Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks. *Proceedings of the twenty-fifth international conference on machine learning* (pp. 792–799). ACM.
Razon, A. R., & Barnden, J. A. (2015). A new approach to automated text readability classification based on concept indexing with integrated part-of-speech *n*-gram

features. *Proceedings of the international conference recent advances in natural language processing. 521*.
Rodriguez, M. B., et al. (2000). Using WordNet to Complement Training Information in Text Categorization. *Proceedings of 2nd International Conference on Recent Advances in Natural Language Processing II: Selected Papers from RANLP'97, 189 of Current Issues in Linguistic Theory (CILT)*, 353–364.
Rostami, M., & Mumivand, H. (2014). Automatic text classification using machine learning algorithm of K-Nearest Neighbor (K-NN) and *N*-gram indexing. *Majlesi Journal of Multimedia Processing, 3*(2), 7–12.
Saiyad, N. Y., Prajapati, H. B., & Dabhi, V. K. (2016). A survey of document clustering using semantic approach. *Proceedings of the international conference on electrical, electronics, and optimization techniques (ICEEOT)*.
Salakhutdinov, R., & Hinton, G. (2009). Deep Boltzmann machines. *Proceedings of the artificial intelligence and statistics* (pp. 448–455). .
Salton, G., & Yang, C. S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation, 29*(4), 11–21.
Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85–117.
Scott, S., & Matwin, S. (1998). Text classification using wordNet hypernyms. *Proceedings of the ACL workshop on usage of wordnet in natural language processing systems* (pp. 45–52). .
Sebastiani, F. (2005). Text categorization. In Alessandro Zanasi (Ed.). *Text mining and its applications* (pp. 109–129). Southampton, UK: WIT Press.
Severyn, A., & Moschitti, A. (2015a). UNITN: Training deep convolutional neural network for Twitter sentiment classification. *Proceedings of the ninth international workshop on semantic evaluation (SemEval 2015)* (pp. 464–469). Association for Computational Linguistics.
Severyn, A., & Moschitti, A. (2015b). Twitter sentiment analysis with deep convolutional neural networks. *Proceedings of the thirty-eigth international ACM SIGIR conference on research and development in information retrieval* (pp. 959–962). ACM.
Siolas, G., & d'Alché-Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. *Proceedings of the international joint conference on neural networks (IJCNN). 5. Proceedings of the international joint conference on neural networks (IJCNN)* (pp. 205–209). IEEE.
Soleimandarabi, M. N., Mirroshandel, S. A., & Sadr, H. A. (2015). Survey of semantic relatedness measures. *International Journal of Computer Science and Network Solutions, 3*(2), 243–247.
Suganya, S., & Gomathi, C. (2013). Syntax and semantics based efficient text classification framework. *International Journal of Computer Applications, 65*(15), 18–21.
Suzuki, M., Tsai, Y. C., Ishida, T., Goto, M., & Hirasawa, S. (2009). Refinement of feature terms and improvement of classification accuracy on multilingual text categorization using character *N*-gram. *Marketing, 11*, 2–95.
Taghipour, K., & Ng, H. T. (2015). Semi-supervised word sense disambiguation using word embeddings in general and specific domains. *Proceedings of the 2015 annual conference of the North American chapter of the association for computational linguistics* (pp. 314–323). .
Tang, D., Qin, B., Liu, T., & Yang, Y. (2015). User modeling with neural network for review rating prediction. *Proceedings of international joint conference on artificial intelligence (IJCAI)* (pp. 1340–1346). .
Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for Twitter sentiment classification. *Proceedings of association for computational linguistics (ACL)* (pp. 1555–1565). .
Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., & Zhou, M. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering, 28*(2), 496–509.
Taşpınar, M., Ganiz, M. C., & Acarman, T. (2017). A feature based simple machine learning approach with word embeddings to named entity recognition on Tweets. *Proceedings of the international conference on applications of natural language to information systems* (pp. 254–259). Springer.
Torunoğlu, D., Telseren, G., Sağtürk, Ö., & Ganiz, M. C. (2013). Wikipedia based semantic smoothing for twitter sentiment classification. *Proceedings of the IEEE international symposium on innovations in intelligent systems and applications (INISTA)* (pp. 1–5). IEEE.
Tsatsaronis, G., Varlamis, I., & Vazirgiannis, M. (2010). Text relatedness based on a word Thesaurus. *Journal of Artificial Intelligence Research, 37*(1), 1–39.
Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. *Proceedings of the forty-eight annual meeting of the association for computational linguistics* (pp. 384–394). Association for Computational Linguistics.
Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the fortieth annual meeting on association for computational linguistics* (pp. 417–424). Association for Computational Linguistics.
Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141–188.
Uysal, A. K., & Gunal, S. (2014). Text classification using genetic algorithm oriented latent semantic features. *Expert Systems with Applications, 41*(13), 5938–5947.
Wang, H., Wang, N., & Yeung, D. Y. (2015). Collaborative deep learning for recommender systems. *Proceedings of the twenty-first ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1235–1244). ACM.
Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using Wikipedia. *Proceeding of the fourteenth 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*(1), 167–174.
Wojtinnek, P., & Pulman, S. (2011). Semantic relatedness from automatically generated semantic networks. *Proceedings of the ninth international conference on computational semantics (IWCS '11)* (pp. 390–394). Association for Computational Linguistics.
Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. *Proceedings of the thirty-second annual meeting on association for computational linguistics (ACL '94)* (pp. 133–138). Association for Computational Linguistics.
Yang, L., Li, C., Ding, Q., & Li, L. (2013). Combining lexical and semantic features for short text classification. *Procedia Computer Science, 22*, 78–86.
Zelikovitz, S., & Hirsh, H. (2004). Transductive LSI for short text classification problems. *Proceedings of Florida artificial intelligence research society conference (FLAIRS)* (pp. 556–561). .
Zesch, T., Muller, C., & Gurevych, I. (2008). Using Wiktionary for computing semantic relatedness. *Proceedings of the twenty-third national conference on artificial intelligence (AAAI'08)* (pp. 861–866). AAAI Press.
Zesch, T., & Gurevych, I. (2010). Wisdom of crowds versus wisdom of linguists – measuring the semantic relatedness of words. *Natural Language Engineering, 16*(1), 25–59.
Zhang, Z., Gentile, A. L., & Ciravegna, F. (2012). Recent advances in methods of lexical semantic relatedness – a survey. *Natural Language Engineering, 1*(1), 1–69.
Zhang, W., Yoshida, T., & Tang, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, 21*(8), 879–886.
Zhang, Z., Gentile, A., & Ciravegna, F. (2011). Harnessing different knowledge sources to measure semantic relatedness under a uniform model. *Proceedings of the conference on empirical methods in natural language processing (EMNLP '11)* (pp. 991–1002). Association for Computational Linguistics.
Zhou, X., Zhang, X., & Hu, X. (2008). Semantic Smoothing for Bayesian text classification with small training data. *Proceedings of the SIAM international conference on data mining (SDM)* (pp. 289–300). .
Zipf, G. K. (1949). *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.
Zubiaga, A. (2012). Enhancing navigation on Wikipedia with social tags. arXiv preprint arXiv:1202.5469.