Hindawi
Complexity
Volume 2018, Article ID 7130146, 10 pages
https://doi.org/10.1155/2018/7130146

Research Article

Deep Learning- and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification

Zeynep H. Kilimci[1] **and Selim Akyokus**[2]

[1] *Computer Engineering Department, Dogus University, Istanbul 34722, Turkey*
[2] *Computer Engineering Department, İstanbul Medipol University, Istanbul 34722, Turkey*

Correspondence should be addressed to Zeynep H. Kilimci; hkilimci@dogus.edu.tr

Received 17 April 2018; Accepted 25 September 2018; Published 9 October 2018

Guest Editor: Ireneusz Czarnowski

The use of ensemble learning, deep learning, and effective document representation methods is currently some of the most common trends to improve the overall accuracy of a text classification/categorization system. Ensemble learning is an approach to raise the overall accuracy of a classification system by utilizing multiple classifiers. Deep learning-based methods provide better results in many applications when compared with the other conventional machine learning algorithms. Word embeddings enable representation of words learned from a corpus as vectors that provide a mapping of words with similar meaning to have similar representation. In this study, we use different document representations with the benefit of word embeddings and an ensemble of base classifiers for text classification. The ensemble of base classifiers includes traditional machine learning algorithms such as naïve Bayes, support vector machine, and random forest and a deep learning-based conventional network classifier. We analysed the classification accuracy of different document representations by employing an ensemble of classifiers on eight different datasets. Experimental results demonstrate that the usage of heterogeneous ensembles together with deep learning methods and word embeddings enhances the classification performance of texts.

1. Introduction

Recently, text classification/categorization has gained remarkable attention of many researchers due to the huge number of documents and text available on the different digital platforms. Given a text or document, the main objective of text classification is to classify the given text into a set of predefined categories by using supervised learning algorithms. The supervised learning algorithms can be trained to generate a model of relationship between features and categories from samples of a dataset. Using the trained model, the learning algorithm can predict the category of a given document.

A text classification task consists of parsing of documents, tokenization, stemming, stop-word removal, and representation of documents in document-term matrix with different weighting methods, feature selection, and selection of the best classifiers by training and testing [1]. Different methods are used for each of the subtasks given above. Each document is usually represented in vector space model (also called bag of words model). In vector space model, each document is represented as a vector that includes a set of terms (words) that appears in the document. The set of documents with their vector representation forms the document-term matrix. The significance of each term in a document is computed by utilizing different term weighting methods. Common term weighting methods include Boolean, term frequency, and TF-IDF weighting schemes. In addition, there has been a recent interest to employ word embeddings to represent documents. There are many types of supervised classifiers used in text categorization. A review and comparison of supervised algorithms are presented in papers [1, 2]. Some of the commonly used supervised algorithms include naïve Bayes (NB), k-nearest neighbours (k-NN), decision trees (DT), artificial neural networks (ANN), and support vector machines (SVM). Deep learning networks have also been attracting attention of researchers in text classification due to their high performance with less need of engineered features.

Another trend in machine learning is to increase the classification performance by using an ensemble of classifiers. In an ensemble system, a group of base classifiers is employed. If different types of classifiers are used as base learners, then such a system is called heterogeneous ensemble, otherwise homogenous ensemble. In this study, we focus on heterogeneous ensembles. An ensemble system is composed of two parts: ensemble generation and ensemble integration [3–7]. In ensemble generation part, a diverse set of models is generated using different base classifiers. Naïve Bayes, support vector machine, random forest, and convolutional neural network learning algorithms are used as base classifiers in this study. There are many integration methods that combine decisions of base classifiers to obtain a final decision [8–11]. For ensemble integration, we used majority voting and stacking methods.

In a previous paper [12], we presented an ensemble of heterogeneous classifiers to enhance the text classification performance. In this study, we try to expand our work by using deep learning methods, which have produced state-of-the-art results in many domains including natural language processing (NLP) and text classification [13]. One of the well-known deep learning-related methods is word embeddings. Word embeddings provide a vector representation of words learned from a corpus. It maps words into a vector of real numbers. While words with similar meaning are mapped into similar vectors, a more efficient representation of words with a much lower dimensional space is obtained when compared with simple bag-of-words approach. Convolutional neural network model (CNN) is another deep learning method employed in this study. CNN is added into our set of base classifiers in order to improve accuracy of the ensemble of heterogeneous classifiers. In addition, we use different document representation methods including TF-IDF weighted document-term matrix, mean of word embeddings, and TF-IDF weighted document matrix enhanced with addition of mean vectors of word embeddings as features. We have performed experiments on eight different datasets in which four of them are in Turkish. In summary, this paper utilizes and analyses an ensemble of classifiers including CNN model with word embeddings and different document representation methods to enhance the performance of text classification.

We have evaluated and discussed the performance of an ensemble of five heterogeneous base learners and two integration methods using three different document representation methods on eight different datasets on this paper. This paper is organized as follows. Section 2 gives related research on text categorization using ensemble systems, word embeddings, and CNN. Section 3 presents base learners and ensemble fusion methods used in experimental studies. Experimental setup and results are given in Section 4. Section 5 summarizes and discusses results and outlines future research directions.

2. Related Work

This section gives a brief summary of ensemble systems, pretrained word embeddings and deep neural networks related to text classification/categorization problem. High dimensionality of input feature space, sparsity of document vectors, and the presence of few irrelevant features are the main characteristics of text categorization problem that differs from other classification problems [14].

The study of Larkey and Croft [15] is one of the early works of applying ensemble systems to text categorization. They used an ensemble of three classifiers, k-nearest neighbour, relevance feedback, and Bayes classifiers to categorize medical documents. Dong and Han [16] used three different variants of naive Bayes and SVM classifiers. They compared the performance of six different homogenous ensembles and a heterogeneous ensemble classifier. Fung et al. [17] use a heterogeneous ensemble classifier that uses a dynamic weighting function to combine decisions. A pairwise ensemble approach is presented in [18] that achieves better performance than popular ensemble approaches bagging and ECOC. Keretna et al. [19] have worked on named entity recognition problem using an ensemble system. In another study done by Gangeh et al. [20], random subspace method is applied to text categorization problem. The paper emphases the estimation of ensemble parameters of size and the dimensionality of each random subspace submitted to the base classifiers. Boroš et al. [21] applied ensemble methods to multiclass text documents where each document can belong to more than one category. They evaluated the performance of ensemble techniques by using multilabel learning algorithms. Elghazel et al. [22] propose a novel ensemble multilabel text categorization algorithm, called multilabel rotation forest (MLRF), based on a combination of rotation forest and latent semantic indexing. Sentiment analysis with ensembles is currently a popular topic among researchers. For Twitter sentiment analysis, an ensemble classifier is proposed in [23] where the dataset includes very short texts. A combination of several polarity classifiers provides an improvement of the base classifiers. In a recent study, the predictive performance of ensemble learning methods on Twitter text documents that are represented by keywords is evaluated by Onan et al. [24] empirically. The five different ensemble methods that use four different base classifiers are applied on the documents represented by keywords.

Representation of text documents with the benefit of word embeddings is another current trend in text classification and natural language processing (NLP). Text representation with word embeddings has been successful in many of NLP applications [13]. Word embeddings are low-dimensional and dense vector representation of words. Word embeddings can be learned from a corpus and can be reused among different applications. Although it is possible to generate your own word embeddings from a dataset, many investigators prefer to use pretrained word embeddings generated from a large corpus. The generation of word embeddings requires a large amount of computational power, preprocessing, and training time [25]. The most commonly used pretrained word embeddings include Word2Vec

[26, 27], GloVe [28], and fastText [29]. In this study, we use pretrained Word2vec embeddings in English and Turkish.

Supervised deep learning networks can model high-level abstractions and provide better classification accuracies compared with other supervised traditional machine algorithms. For this reason, supervised deep learning architectures have received significant attention in NLP and text classification recently. Kalchbrenner et al. [30] described a dynamic convolutional network that uses dynamic k-max pooling, a global pooling operation over linear sequences for sentiment classification of movie reviews and Twitter. Kim [31] reports a series of experiments with four different convolutional neural networks for sentence-level sentiment classification tasks using pretrained word vectors. Kim achieves good performance results on different datasets and suggests that the pretrained word vectors are universal and can be used for various classification tasks. In [32], a new deep convolutional neural network is proposed to utilize from character-level to sentence-level information to implement sentiment classification of short texts. In [33, 34], text classification is realized with a convolutional neural network that accepts a sequence of encoded characters as input rather than words. It is shown that character-level coding is an effective method. Johnson and Zhang [35] propose a semisupervised convolutional network for text classification that learns embeddings of small text regions. Joulin et al. [29] proposes a simple and efficient baseline classifier that performs as well as deep learning classifiers in terms of accuracy and runs faster. Conneau et al. [36] present a new architecture called very deep (VDCNN) for text processing which operates directly at the character level and uses only small convolutions and pooling operations. Kowsari et al. [37] introduce a new approach to hierarchical document classification, called HDLTex that employs multiple deep learning approaches to produce hierarchical classifications.

3. Ensemble Learning, Word Embeddings, and Representations

This section gives a summary of learning algorithms, ensemble integration techniques, word embeddings, and document representation methods used in this study.

3.1. Base Learners. In this study, we employ multivariate Bernoulli naïve Bayes (MVNB), multinomial naïve Bayes (MNB), support vector machine (SVM), random forest (RF), and convolutional neural network (CNN) learning algorithms as base classifiers to generate a heterogeneous ensemble system.

3.1.1. MVNB and MNB. As a simple probabilistic classifier, naïve Bayes is based on Bayes' theorem with the assumption of independence of features from each other. There are two types of naïve Bayes classifier frequently used for text categorization: multivariate Bernoulli naïve Bayes (MVNB) and multinomial naïve Bayes (MNB). In MVNB, each document is represented by a vector with binary variables that can take values 1 or 0 depending upon the presence of a word in the document. In MNB, each document vector is represented by the frequency of words that appear in the document. Equation (1) defines MVNB classifier with Laplace smoothing. The occurrence of the term t in document i is indicated by B_{it} which can be either 1 or 0. $|D|$ indicates the number of labelled training documents. $P(c_j \mid d_i)$ is 1 if document i is in class j. The probability of term w_t in class c_j is as follows [38]:

$$P\left(w_t|c_j\right) = \frac{1 + \sum_{i=1}^{D} B_{it} P\left(c_j|d_i\right)}{2 + \sum_{i=1}^{D} P\left(c_j|d_i\right)}. \tag{1}$$

3.1.2. SVM. Support vector machine (SVM) is binary classifier that divides an n-dimensional space with n features into two regions related with two classes [39]. The n-dimensional hyperplane separates two regions in a way that the hyperplane has the largest distance from training vectors of two classes called support vectors. SVM can also be used for a nonlinear classification using a method called the kernel trick that implicitly maps input instances into high-dimensional feature spaces that can be separated linearly. In SVM, the use of different kernel functions enables the construction of a set of diverse classifiers with different decision boundaries.

3.1.3. RF. Random forests (RF) are a collection of decision tree classifiers introduced by Breiman [40]. It is a particular implementation of bagging in which decision trees are used as base classifiers. Given a standard training set, the bagging method generates a new training set by sampling with replacement for each of base classifiers. In standard decision trees, each node of the tree is split by the best feature among all other features using splitting criteria. To provide randomness at feature space, random forest algorithm first selects a random subset of features and then decides on the best split among the randomly selected subset of features. Random forests are strong against overfitting because of randomness applied in both sample and feature spaces.

3.1.4. CNN. Convolutional neural networks (CNNs) are a class of deep learning networks that have achieved remarkable results in image recognition and classification [41–43]. CNN models have subsequently been shown to be effective for natural language processing tasks [13]. CNN is a feed-forward network with an input and output layer and hidden layers. Hidden layers consist of convolutional layers interleaved with pooling layers. The most important block of CNN is the convolutional layer. Convolutional layer applies a convolution filter to input data to produce a feature map to combine information with data on the filter. Multiple filters are applied to input data to get a stack of feature maps that becomes the final output of the convolutional layer. The values of filters are learned during the training process. Convolution operation captures local dependencies or semantics in the regions of original data. An additional activation function (usually RELU (rectified linear unit)) is applied to feature maps to add nonlinearity to CNN. After convolution, a pooling layer reduces the number of samples in each feature map and retains the most important information. The pooling layer shortens the training time and reduces the dimensionality of data and overfitting. Max pooling is the most common type of pooling function that takes the largest value in a specified neighbourhood window. CNN architectures consist of a series of convolutional layers interleaved with pooling layers, followed by a number of fully connected layers.

CNN is originally applied on image processing and recognition tasks where input data is in a two-dimensional (2D) structure. On image processing, CNN exploits spatial local correlation of 2D images, learns, and responds to small regions of 2D images. In natural language processing and text classification, words in a text or document are converted into a vector of numbers that has one-dimensional (1D) structure. On text processing applications, CNN can exploit the 1D structure of data (word order) by learning small text regions of data in addition to learning from a bag of independent word vectors that represents an entire document [44].

3.2. Ensemble Integration Strategies. To combine decisions of individual base leaners MVNB, MNB, SVM, RF, and CNN, majority voting and stacking methods are used in this study. In majority voting method, an unlabelled instance is classified according to the class that obtains the highest number of votes from collection of base classifiers. In stacking method, also called stacked generalization, a metalevel classifier is used to combine the decision of base-level classifiers [45]. The stacking method consists of two steps. In the first step, a set of base-level classifiers $C_1, C_2, \dots, C_n$ is generated from a sample training set S that consists of feature examples $s_i = (\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ is feature vector and y_i is prediction (class label). A meta-dataset is constructed from the decisions of base-level classifiers. The meta-dataset contains an instance for predictions of classifiers in the original training dataset. The meta-dataset is in the form of $m_i = (d_i, y_i)$ where d_i is the prediction of individual n base classifiers. The meta-dataset can also include both original training examples and decisions of base-level classifiers in the form of $m_i = (\mathbf{x}_i, d_i, y_i)$ to improve performance. After the generation of meta-dataset, a metalevel classifier is trained with meta-dataset and used to make predictions. In our study, the meta-dataset includes both original training examples and decisions of base-level classifiers.

3.3. Word Embeddings with Word2vec. Word embeddings are an active research area that tries to discover better word representations of words in a document collection (corpus). The idea behind all of the word embeddings is to capture as much contextual, semantical, and syntactical information as possible from documents from a corpus. Word embeddings are a distributed representation of words where each word is represented as real-valued vectors in a predefined vector space. Distributed representation is based on the notion of distributional hypothesis in which words with similar meaning occur in similar contexts or textual vicinity. Distributed vector representation has proven to be useful in many natural language processing applications such as named entity recognition, word sense disambiguation, machine translation, and parsing [13].

Currently, Word2vec is the most popular word embedding technique proposed by Mikolov et al. [26, 27]. The Word2vec method learns vector representations of words from a training corpus by using neural networks. It maps the words that have similar meaning into vectors that will be close to each other in the embedded vector space. Word2vec offers a combination of two methods: CBOW (continuous bag of words) and skip-gram model. While the CBOW model predicts a word in a given context, the Skip-gram model predicts the context of a given word. Word2vec extracts continuous vector representations of words from usually very large datasets. While it is possible to generate your own vector representation model from a given corpus, many studies prefers to use pretrained models because of high computational power and training time required for large corpuses. The pretrained models have been found useful in many NLP applications.

3.4. Document Representation Methods. The effective representations of documents in a document collection have a significant role in the success performance of text processing applications. In many text classification applications, each document in a corpus is represented as a vector of real numbers. Elements of a vector usually correspond to terms (words) appearing in a document. The set of documents with their vector representation forms document-term matrix. The significance of each term in a document is computed by using different term weighting methods. Traditional term weighting methods include Boolean, term frequency, and TF-IDF weighting schemes. TF-IDF is the common weighting method used for text processing. In this representation, the term frequency for each word is multiplied by the inverse document frequency (IDF). This reduces the importance of common terms in the collection and also increases the influence of rare words which have relatively low frequencies. As explained above, word embedding is also another effective method to represent documents as a vector of numbers.

In this study, we use and compare the following document representation methods:

(i) TF-IDF. A document vector consists of words appearing in a document weighted with TF-IDF scheme

(ii) Avg-Word2vec. A document vector is obtained by taking average of all vectors of word embeddings appearing in a document by using pretrained models

(iii) TF-IDF + Avg-Word2vec. A document vector includes both TF-IDF and Avg-Word2vec vectors

4. Experiments

In this study, we have evaluated the performance of an ensemble of eight heterogeneous base learners with two integration methods using three different document representation methods on eight different datasets.

4.1. Datasets. We use eight different datasets with different sizes and properties to explore the classification performances of the heterogeneous classifier ensembles in Turkish and English. Turkish is an agglutinative language in which words are composed of a sequence of morphemes (meaningful word elements). A single Turkish word can correspond to a sentence that can be expressed with several words in other languages. Turkish datasets include news articles from Milliyet, Hurriyet, 1150haber, and Aahaber datasets. English datasets consist of 20News-19997, 20News-18828, Mini-news, and WebKB4. The properties of each dataset are explained below.

Milliyet dataset includes text from columns of Turkish newspaper Milliyet from years 2002 to 2011. It contains 9 categories and 1000 documents for each category. The categories of this dataset are café (cafe), dünya (world), ege (region), ekonomi (economy), güncel (current), siyaset (politics), spor (sports), Türkiye (Turkey), and yaşam (life). *Hurriyet* dataset includes news from 2010 to 2011 on Turkish newspaper Hurriyet. It contains six categories and 1000 documents for each category. Categories in this dataset are dünya (world), ekonomi (economy), güncel (current), spor (sports), siyaset (politics), and yaşam (life). *1150haber* dataset is obtained from a study done by Amasyalı and Beken [46]. It consists of 1150 Turkish news texts in five classes (economy, magazine, health, politics, and sports) and 230 documents for each category. *Aahaber*, collected by Tantug [47], is a dataset that consists of newspaper articles broadcasted by Turkish National News Agency, Anadolu Agency. This dataset includes eight categories and 2500 documents for each category. Categories are Turkey, world, politics, economics, sports, education science, "culture and art", and "environment and health".

Milliyet and 1150haber include the writings of the column writers; therefore, they are longer and more formal. On the other hand, Hurriyet and Aahaber datasets contain traditional news articles. They are more irregular, much shorter than documents of the other datasets.

The 20 newsgroup is a popular English dataset used in many text classification and clustering experiments [48]. The 20 newsgroup dataset is a collection of about 20,000 documents extracted from 20 different newsgroups. The data is almost evenly partitioned into 20 categories. We use three versions of newsgroup dataset. The first one is the original 20 newsgroup dataset that includes 19,997 documents. It is called as *20News-19997*. The second one is named *20News-18828* with 18,828 documents, and it covers less number of documents than the original dataset. This dataset includes messages with only "From" and "Subject" headers with the removal of cross-post duplicate messages. The third one is a small subset of the original dataset composed of 100 postings per class, and it is called as *mini-newsgroups*. The last dataset is called *WebKB* [49] which includes web pages gathered from computer science departments of different universities. These web pages are composed of seven categories (student, faculty, staff, course, project, department, and other) and contain approximately 8300 pages. Another version of WebKB is called *WebKB4* where the number of categories is reduced to four. We use WebKB4 in our experiments.

Table 1: Number of classes ($|C|$), documents ($|D|$), and words ($|V|$) in each document.

Dataset	$\|C\|$	$\|D\|$	$\|V\|$
20News-18828	20	18,828	50,570
20News-19997	20	19,997	43,553
Mini-news	20	2000	13,943
WebKB4	4	4199	16,116
1150haber	5	1150	11,040
Milliyet	9	9000	63,371
Hurriyet	8	6000	18,280
Aahaber	8	2000	14,396

Characteristics of the datasets without application of any preprocessing procedures are given in Table 1 where $|C|$ is the number of classes, $|D|$ is the number of documents, and $|V|$ is the vocabulary size. We only filter infrequent terms whose document frequency is less than three. We do not apply any stemming or stop-word filtering in order to avoid any bias that can be introduced by stemming algorithms or stop-word lists.

4.2. Experiment Results. We use repeated holdout method in our experiments. We randomly divide a dataset into two halves where 80% of data is used for training and 20% for testing. To get a reliable estimation, we repeat the holdout process 10 times and an overall accuracy is computed by taking averages of each iteration. Classification accuracies and computation times of algorithms on different datasets are presented below.

As a first step, we calculate accuracies of individual classifiers to compare and observe results that we obtain by using an ensemble of classifiers with different representation methods. As explained before, base classifiers include MVNB, MNB, SVM, RF, and CNN. Table 2 lists the accuracies of these classifiers on eight different Turkish and English datasets. The best accuracy is obtained for each dataset and is shown in boldface. As it can be seen from Table 2, there is no best algorithm that performs well on each dataset like in many machine language problems. It seems that RF and CNN generally produce better accuracies from the rest of the algorithms. The random forest (RF) is the best algorithm in our experiments. This might be due to ensemble strategy applied in RF algorithm that uses a set of decision trees for classification. CNN also performs well. That might be because of the use of different convolutional filters that work like an ensemble system that extracts different features from datasets. The order of average classification accuracies of single classifiers can be summarized as follows: RF > CNN > MNB > SVM > MVNB.

To construct a heterogeneous ensemble system, we use MVNB, MNB, SVM, RF, and CNN algorithms as base classifiers. The decisions of each of these base classifiers are combined with majority voting (MV) and stacking (STCK) integration methods as described in Section 3.2. Each dataset is represented with traditional TF-IDF weighting scheme. Table 3 demonstrates the classification accuracies of heterogeneous ensemble systems with majority voting (Heter-MV) and stacking (Heter-STCK) together

Table 2: Classification accuracies of single classifiers on datasets represented with TF-IDF weighting scheme.

Dataset	MVNB	MNB	SVM	RF	CNN
20News-18828	75.89	91.49	**91.50**	90.36	89.23
20News-19997	63.91	**81.29**	63.14	78.56	75.56
Mini-news	77.78	82.93	**92.40**	90.44	91.7
WebKB4	79.30	86.74	91.15	89.75	**91.56**
1150haber	85.17	**95.30**	92.52	95.07	94.37
Milliyet	83.19	83.91	92.75	93.05	78.06
Hurriyet	77.14	82.29	81.44	85.11	**87.31**
Aahaber	82.06	83.26	80.89	88.26	**90.19**
Average	78.06	85.90	85.72	**88.83**	87.25
Computation time	1 h 13 m	1 h 45 m	2 h 17 m	3 h 34 m	5 h 23 m

Table 3: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with TF-IDF weighting.

Methods	20News-18828	20News-19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	75.89	63.91	77.78	79.30	85.17	83.19	77.14	82.06	1 h 13 m
MNB	91.49	81.29	82.93	86.74	95.30	83.91	82.29	83.26	1 h 45 m
SVM	91.50	63.14	92.40	91.15	92.52	92.75	81.44	80.89	2 h 17 m
RF	90.36	78.56	90.44	89.75	95.07	93.05	85.11	88.26	3 h 34 m
CNN	89.23	75.56	91.70	91.56	94.37	78.06	87.31	90.19	5 h 23 m
Heter-MV	92.39	79.63	92.60	90.85	94.98	92.34	88.41	89.94	7 h 55 m
Heter-Stck	93.25	81.29	94.07	93.57	96.77	94.09	89.55	92.70	8 h 31 m

with each of the single classifiers. As it can be seen from Table 3, an ensemble system with stacking integration strategy always performs better than single classifiers and the ensemble model with majority integration strategy. On our previous study [50], we obtain classification accuracies 85.44 and 87.73 for Turkish Hurriyet and Aahaber datasets, respectively, represented with TF (term frequency) weighting by using an ensemble of classifiers without CNN. With the inclusion of CNN to our set of base learners, improved accuracies are obtained with 89.55 and 92.70 for Turkish Hurriyet and Aahaber datasets, respectively.

In text mining applications, different words appearing in a document collection form feature set where the number of features is usually expressed in thousands. High dimensionality of feature space is a problem in text classification when documents are represented with "bag of words" model. Word2vec can be used as a feature extraction technique to reduce the number of features. The average of Word2vec vectors of words is employed to represent documents. Given a document d represented with n words $d = w_1, w_2, \ldots, w_n$, words appearing in a document are represented with Word2vec embedding vectors $e_{w_1}, e_{w_2}, \ldots, e_{w_n}$ by looking up the vector representation of a word from a pretrained embedding model.

Each document is represented by taking average of word embeddings as follows:

$$e_d = \frac{1}{n}\sum_{i=1}^{n} e_{w_i}. \tag{2}$$

Google has used Google News dataset that contains about 100 billion words to obtain pretrained vectors with the Word2vec skip-gram algorithm [26, 51]. The pretrained model includes word vectors for about 3 million words and phrases. Each vector has 300 dimensions or features. A pretrained Turkish Word2vec model is constructed with all Wikipedia articles written in Turkish [52]. We use these pretrained models in English and Turkish to represent documents with 300 dimensions or features.

Table 4 shows classification accuracies of documents represented by average of Word2vec vectors, called as Avg-Word2vec, in this study. Classification accuracies of single classifiers and heterogeneous ensemble systems are given in Table 4. Ensemble method with stacking integration strategy (Heter-Stck) produces better outcomes than the ensemble method with majority voting integration strategy (Heter-MV). We observe that there is a slight decrease in the classification accuracies of documents represented with Avg-Word2vec on the six datasets (20News-18828, 20News-19997, Mini-news, 1150-haber, Milliyet, and Hurriyet) when compared with the classification accuracies (Heter-Stck) of documents represented with TF-IDF given in Table 3. On the contrary, there is a slight improvement on two datasets (WebKB4, Aahaber). Avg-Word2vec-based classification performance of the system is competitive although a much less number of features are employed.

In the last experiment, each document is represented by concatenation of TF-IDF and average of Word2vec vectors. Although the size of the feature set is increased, we observe better classification accuracies than the other representation

Table 4: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with the average of Word2vec vectors.

Methods	20News 18828	20News 19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	74.23	61.01	76.18	79.77	87.27	81.12	77.58	83.11	56 m
MNB	89.50	80.44	81.67	87.05	96.13	81.55	82.39	84.36	1 h 17 m
SVM	88.51	62.78	90.35	91.71	93.89	89.27	82.00	82.23	1 h 50 m
RF	87.34	76.65	89.27	90.06	97.56	90.30	84.45	89.78	2 h 44 m
CNN	90.17	76.91	92.55	92.22	95.14	76.10	88.07	93.26	4 h 55 m
Heter-MV	89.62	78.20	91.45	93.45	95.81	90.93	87.18	92.78	6 h 30 m
Heter-Stck	90.85	80.94	93.78	94.56	96.34	92.05	88.96	93.80	7 h 23 m

Table 5: The list of classification accuracies of single classifiers and heterogeneous ensemble systems with majority voting and stacking integration strategies on datasets represented with TF-IDF and the average of Word2vec vectors.

Methods	20News-18828	20News-19997	Mini-news	Web KB4	1150haber	Milliyet	Hurriyet	Aahaber	Computation time
MVNB	75.83	63.78	77.00	80.21	87.97	83.70	78.80	83.71	1 h 57 m
MNB	90.27	81.74	81.98	87.80	96.78	84.42	83.09	84.80	2 h 24 m
SVM	91.51	64.08	92.35	92.05	94.20	91.36	83.34	81.56	2 h 55 m
RF	91.25	79.15	91.47	90.92	97.99	92.10	85.92	90.38	4 h 5 m
CNN	91.88	77.80	92.12	93.01	95.59	79.49	89.27	94.15	5 h 48 m
Heter-MV	92.90	79.90	92.68	93.75	96.00	92.13	88.53	93.10	8 h 32 m
Heter-Stck	94.30	82.23	95.06	95.08	97.40	93.90	90.02	94.86	9 h 20 m

methods. Table 5 gives the classification accuracies of ensemble systems together with the base classifiers. Heterogeneous ensemble with stacking integration strategy (Heter-Stck) presents the best classification accuracies among the other experiments. When compared with the classification accuracies of documents represented with TF-IDF given in Table 2, we obtain better classification accuracies on seven of the datasets except Milliyet in which accuracy is decreased from 94.09 to 93.9 (0.2%). It can be concluded that the inclusion of Avg-Word2vec vectors to TF-IDF vectors as additional features boosts classification success of the ensemble system.

It is difficult to compare the performance of our results with other studies because of the lack of works with similar combinations of different datasets, representation models, and ensemble approaches. Although a comparison of baseline classifiers and heterogeneous ensemble systems with different representation techniques is given in this study, we also report the classification accuracies of a number of research works here. In [53], Pappagari et al. propose an end-to-end multiscale CNN framework for topic identification by employing 20 newsgroups and Fisher datasets. The classification success of our proposed system outperforms with 94.3% while the accuracy performance of the work [53] presents 86.1% classification accuracy.

In another study [54], a graph-based framework is presented by utilizing SVM for text classification problem. Two data representation models (TF-IDF and Word2vec) and datasets (20 newsgroups and WebKB) are studied in this work [54]. 20 newsgroup dataset with TF-IDF representation exhibits 83.0% classification accuracy while the Word2vec version of the same dataset presents 75.8% classification success in [54]. The TF-IDF representation model of our study performs 91.5% and the Word2vec representation of our work achieves 88.5% classification accuracies for 20 newsgroup dataset. Moreover, their study [54] with WebKB dataset and TF-IDF representation exhibits 89.9% classification accuracy while the Word2vec version of the same dataset presents 86.6% classification success. The TF-IDF representation model of our work performs 91.2%, and the Word2vec representation of ours represents 91.7% classification accuracies for WebKB dataset. In [55], Zheng et al. propose a bidirectional hierarchy skip-gram model to mine topic information within given texts. They use CNN and SVM as classification algorithms and 20 newsgroups and WebKB as datasets like in our study. Our proposed system performs 91.9% classification success with CNN algorithm while their CNN implementation exhibits 79.2% for 20 newsgroups in [55]. The proposed system of our study performs 91.5% classification success with SVM algorithm while SVM exhibits 85.9% for 20 newsgroups in [55]. For the WebKB dataset, CNN implementation of our work performs 93.0% classification success while their CNN method exhibits 91.8%. With SVM algorithm on WebKB dataset, we obtain 92.1% classification success while their SVM implementation exhibits 88.1%. The proposed heterogeneous ensemble systems given in this study always perform well compared with other studies reported above in terms of classification accuracies.

5. Conclusion

In this paper, we focus to enhance the overall accuracy of a text classification system by using ensemble learning, deep learning, and effective document representation methods. It is known that the classifier ensembles boost the overall classification performance by depending on two factors, namely, individual success of the base learners and the diversity of them. The different learning algorithms, namely, variants of two naïve Bayes (MVNB and MNB), support vector machine (SVM), random forest (RF), and currently popular convolutional neural networks (CNNs), are chosen to provide diversity for the ensemble system. The majority voting and stacking ensemble integration strategies are performed to consolidate the final decision of the ensemble system. Word embeddings are also utilized to raise the overall accuracy of text classification. Word embeddings can capture contextual, semantical, and syntactical information in a textual vicinity from documents from a corpus.

A set of experiments is performed on eight different datasets represented with different methods using an ensemble of classifiers MVNB, MNB, SVM, RF, and CNN. As a result of experiments, some of the main findings of this study are as follows:

(i) RF and CNN are the best performing single classifiers among others. The order of classification accuracies of single classifiers can be summarized as RF > CNN > MNB > SVM > MVNB

(ii) An ensemble of classifiers increases the classification accuracies of texts on different datasets that have different characteristics and distributions

(iii) A set of heterogeneous ensemble of classifiers can provide slight performance increases in terms of accuracy when compared with homogenous ensemble of classifiers [12, 50]

(iv) Stacking is a better ensemble integration strategy than majority voting

(v) The inclusion of state-of-art deep learning CNN classifier to the set of classifiers of an ensemble system can provide further enhancement

(vi) The use of pretrained word embeddings is an effective method to represent documents. It can be a good feature reduction method without losing much in terms of classification accuracy

(vii) Inclusion of word embeddings to TF-IDF weighted vectors as additional features provides a further improvement in text classification because word embeddings can capture contextual, semantical, and syntactical information from text

In the future, we plan to use different pretrained word embedding models, document representation methods using word embeddings, and other deep learning algorithms for text classification and natural language processing tasks.

Data Availability

We used publicly available datasets in our experiments. If necessary, we can share those datasets.

Disclosure

This work is prepared after invitation of the paper [12] published in 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[2] C. C. Aggarwal and C. X. ZhaiC. C. Aggarwal and C. X. Zhai, “A survey of text classification algorithms,” in *Mining Text Data*, pp. 163–222, Springer, 2012.

[3] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[4] R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[5] S. Reid, “A review of heterogeneous ensemble methods,” in *Department of Computer Science*, University of Colorado at Boulder, 2007.

[6] D. Gopika and B. Azhagusundari, “An analysis on ensemble methods in classification tasks,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 7, pp. 7423–7427, 2014.

[7] Y. Ren, L. Zhang, and P. N. Suganthan, “Ensemble classification and regression-recent developments, applications and future directions [review article],” *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[8] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, “A survey of decision fusion and feature fusion strategies for pattern classification,” *IETE Technical Review*, vol. 27, no. 4, pp. 293–307, 2010.

[9] M. Woźniak, M. Graña, and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” *Information Fusion*, vol. 16, pp. 3–17, 2014.

[10] G. Tsoumakas, L. Angelis, and I. Vlahavas, “Selective fusion of heterogeneous classifiers,” *Intelligent Data Analysis*, vol. 9, no. 6, pp. 511–525, 2005.

[11] A. Güran, M. Uysal, Y. Ekinci, and B. Güran, “An additive FAHP based sentence score function for text summarization,” *Information Technology And Control*, vol. 46, no. 1, 2017.

[12] Z. H. Kilimci, S. Akyokus, and S. İ. Omurca, “The evaluation of heterogeneous classifier ensembles for Turkish texts,” in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 307–311, Gdynia, 2017.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” 2018, http://arxiv.org/abs/1708.02709v5.

[14] T. Joachims, “Text categorization with support vector machines: learning with many relevant features,” in *Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, C. Nédellec and C. Rouveirol, Eds., vol. 1398, Springer, Berlin, Heidelberg, 1998.

[15] L. S. Larkey and W. Bruce Croft, "Combining classifiers in text categorization," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '96*, pp. 289–297, Zurich, Switzerland, 1996.

[16] Y.-S. Dong and K.-S. Han, "A comparison of several ensemble methods for text categorization," in *IEEE International Conference onServices Computing, 2004. (SCC 2004). Proceedings. 2004*, pp. 419–422, Shanghai, China, 2004.

[17] G. P. C. Fung, Y. Jeffrey Xu, H. Wang, D. W. Cheung, and H. Liu, "A balanced ensemble approach to weighting classifiers for text classification," in *Sixth International Conference on Data Mining (ICDM'06)*, pp. 869–873, Hong Kong, 2006.

[18] Y. Liu, J. Carbonell, and R. Jin, "A new pairwise ensemble approach for text classification," in *Machine Learning: ECML 2003. ECML 2003. Lecture Notes in Computer Science*, pp. 277–288, Springer, Berlin, Heidelberg, 2003.

[19] S. Keretna, C. P. Lim, D. Creighton, and K. B. Shaban, "Classification ensemble to improve medical named entity recognition," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2630–2636, San Diego, CA, USA, October 2014.

[20] M. J. Gangeh, M. S. Kamel, and R. P. W. Duin, "Random subspace method in text categorization," in *2010 20th International Conference on Pattern Recognition*, pp. 2049–2052, Istanbul, Turkey, August 2010.

[21] M. Boroš, Franky, and J. Maršík, "Multi-label text classification via ensemble techniques," *International Journal of Computer and Communication Engineering*, vol. 1, no. 1, pp. 62–65, 2012.

[22] H. Elghazel, A. Aussem, O. Gharroudi, and W. Saadaoui, "Ensemble multi-label text categorization based on rotation forest and latent semantic indexing," *Expert Systems with Applications*, vol. 57, no. 15, pp. 1–11, 2016.

[23] M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of ensemble methods on Twitter sentiment analysis using NLP techniques," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169-170, Anaheim, CA, February 2015.

[24] A. Onan, S. Korukoglu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.

[25] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Improving the accuracy of pre-trained word embeddings for sentiment analysis," 2017, http://arxiv.org/abs/1711.08609v1.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, http://arxiv.org/abs/1301.3781.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, Advances in Neural Information Processing Systems Conference (NIPS 2013), 2013.

[28] J. Pennington, R. Socher, and C. Manning, "GloVe: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, http://arxiv.org/abs/1612.03651.

[30] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–666, Baltimore, MD, USA, 2014.

[31] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.

[32] C. N. D. Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *The 25th International Conference on Computational Linguistics*, pp. 69–78, Dublin, Ireland, August 2014.

[33] X. Zhang and Y. LeCun, "Text understanding from scratch," 2015, http://arxiv.org/abs/1502.01710.

[34] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, pp. 649–657, Montreal, Canada, 2015.

[35] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in Neural Information Processing Systems*, pp. 919–927, Montreal, Canada, 2015.

[36] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," 2017, http://arxiv.org/abs/1606.01781v2.

[37] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: hierarchical deep learning for text classification," http://arxiv.org/abs/1709.08267v2.

[38] A. McCallum and K. A. Nigam, "Comparison of event models for naive Bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, AAAI Press, Wisconsin, USA, 1998.

[39] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[40] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[44] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, CO, USA, 2015.

[45] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine Learning*, vol. 54, no. 3, pp. 255–273, 2004.

[46] M. F. Amasyalı and A. Beken, "Türkçe kelimelerin anlamsal benzerliklerinin ölçülmesi ve metin sınıflandırmada kullanılması," in *IEEE signal processing and communications applications conference*, Antalya, Turkey, 2009.

[47] A. C. Tantug, "Document categorization with modified statistical language models for agglutinative languages," *International Journal of Computational Intelligence Systems*, vol. 3, no. 5, pp. 632–645, 2010.

[48] https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html.

[49] M. Craven, D. DiPasquo, D. Freitag et al., “Learning to extract symbolic knowledge from the world wide web,” in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. Menlo Park: American Association for Artificial Intelligence*, pp. 509–516, Menlo Park, CA, USA, 1998.

[50] Z. H. Kilimci, S. Akyokus, and S. I. Omurca, “The effectiveness of homogenous ensemble classifiers for Turkish and English texts,” in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–7, Sinaia, Romania, August 2016.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “English pre-trained Word2vec model,” https://code.google.com/archive/p/word2vec/.

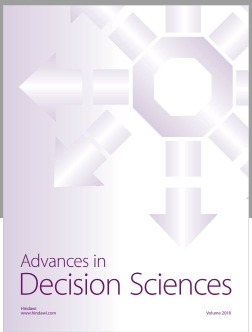
[52] A. Koksal, “Turkish pre-trained Word2vec model,” https://github.com/akoksal/Turkish-Word2Vec.

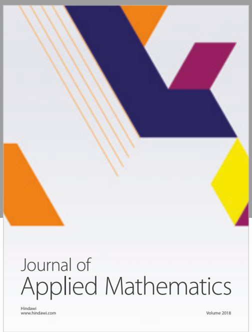
[53] R. Pappagari, J. Villalba, and N. Dehak, “Joint verificationidentification in end-to-end multi-scale cnn framework for topic identification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, April 2018.

[54] K. Skianis, F. Malliaros, and M. Vazirgiannis, “Fusing document, collection and label graph-based representations with word embeddings for text classification,” in *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, New Orleans, Louisiana, United States, June 2018.

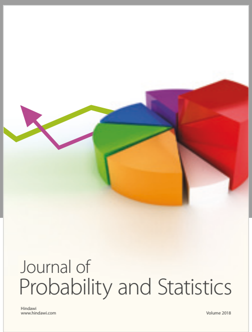
[55] S. Zheng, J. X. Hongyun Bao, Y. Hao, Z. Qi, and H. Hao, “A bidirectional hierarchical skip-gram model for text topic embedding,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 855–862, Vancouver, BC, July 2016.

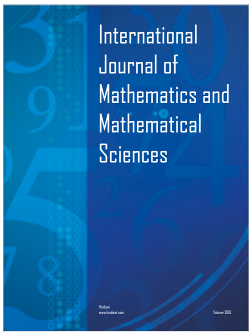
Advances in
Operations Research


Advances in
Decision Sciences

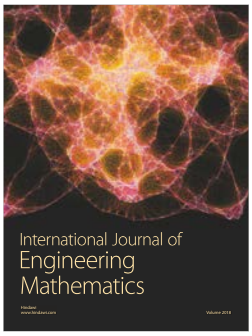

Journal of
Applied Mathematics

The Scientific
World Journal

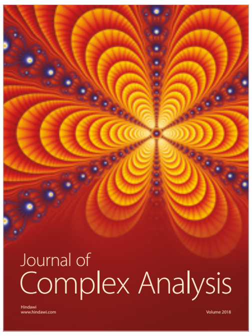

Journal of
Probability and Statistics

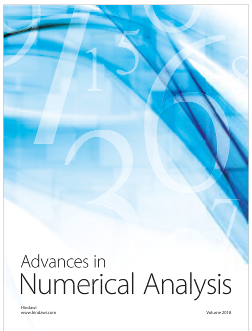

International
Journal of
Mathematics and
Mathematical
Sciences

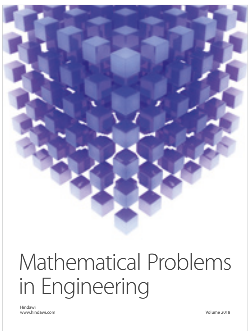
Journal of
Optimization

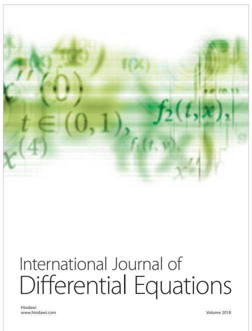

International Journal of
Engineering
Mathematics

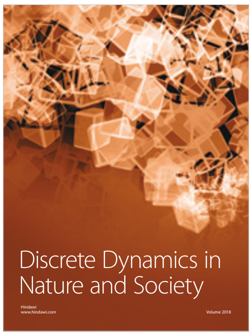
International Journal of
Analysis

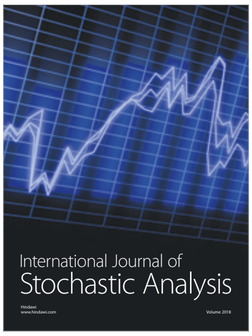

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

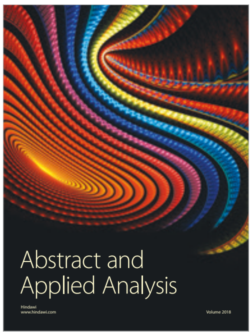

International Journal of
Differential Equations

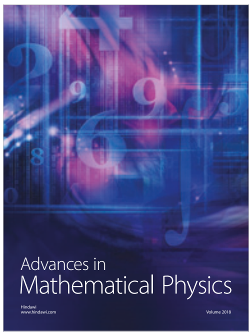

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics