Manuscript Number:

Title: A new document vector representation using word2vec and latent dirichlet allocation

Article Type: Research paper

Keywords: Natural language processing; Text Classification; Word embeddings; Topic modeling; LDA; Doc2vec

Corresponding Author: Mr. Halil İbrahim ÇELENLİ,

Corresponding Author's Institution:
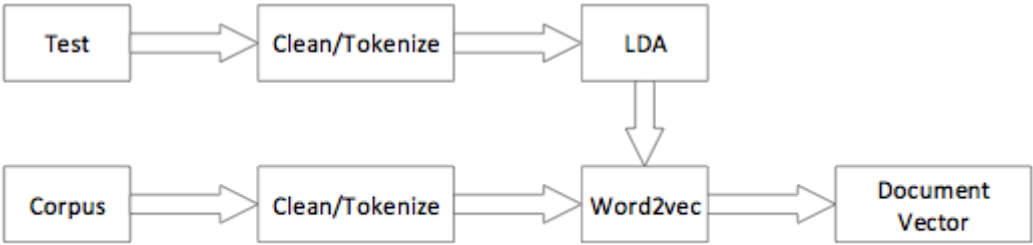
First Author: Halil İbrahim ÇELENLİ

Order of Authors: Halil İbrahim ÇELENLİ; Sevinç İlhan OMURCA; Murat Can GANİZ

Abstract: Word embedding models typically learn a vector representation for each word, based on the co-occurrence statistics in the contexts across a text corpus. On the other hand, topic modeling techniques produce a distribution over words for each latent topic and a distribution over latent topics for each document in a given corpus. We combine word embedding models with topic models for a better semantic representation which in turn used to represent documents. To demonstrate the value of our approach we choose document categorization as the mainstream NLP task. We compare the classification performance of well-known machine learning classifiers on the proposed document vectors and traditional Doc2vec approach. We show that the proposed document vectors; combination of Word2vec and LDA improves the classification results and reduce the complexity of the resulting model.

Graphical Abstract

A new document vector representation using word2vec and latent dirichlet allocation

Halil İbrahim Çelenli,Sevinç İlhan Omurca,Murat Can Ganiz

*Corpus = Test + Train

A new document vector representation using word2vec and latent dirichlet allocation

Halil İbrahim Çelenli[a,*], Sevinç İlhan Omurca[a] and Murat Can Ganiz[b]

[a]*Department of Computer Engineering, Faculty of Engineering, Kocaeli University, Kocaeli, Turkey*
[b]*Department of Computer Engineering, Faculty of Engineering, Marmara University, İstanbul, Turkey*

ARTICLE INFO

Keywords:
Natural language processing
Text Classification
Word embeddings
Topic modeling
LDA
Doc2vec

ABSTRACT

Word embedding models typically learn a vector representation for each word, based on the co-occurrence statistics in the contexts across a text corpus. On the other hand, topic modeling techniques produce a distribution over words for each latent topic and a distribution over latent topics for each document in a given corpus. We combine word embedding models with topic models for a better semantic representation which in turn used to represent documents. To demonstrate the value of our approach we choose document categorization as the mainstream NLP task. We compare the classification performance of well-known machine learning classifiers on the proposed document vectors and traditional Doc2vec approach. We show that the proposed document vectors; combination of Word2vec and LDA improves the classification results and reduce the complexity of the resulting model.

1. Introduction

Since the natural language; text documents are unstructured they need to be converted into structured format such as vectors or matrices. This preprocessing stage; the representation of words or documents is crucial because this process may have a fundamental effect on the performance of the Natural Language Processing (NLP) tasks such as document classification. In this study we focus on creating a novel document representations by combining word embedding models with topic models. We evaluate the value of this approach in document classification domain. The aim of document classification is to assign the most appropriate label to a document from a set of predetermined labels. Many document classification approaches use bag-of-words (BoW) representations which results in usually very high dimensional and sparse vectors. Especially the sparsity creates several problems for machine learning algorithms that operate on these representations. They ignore the semantic relationships between different words and therefore the documents (Jiang et al., 2018). The generalization power of the classification models become stronger when they make use of the semantic information.

Recently, word embedding techniques widely used to represent words as low dimensional dense vectors. These methods are capable of implicitly incorporating both semantic and syntactic similarity information into these vector representations because they employ context or co-occurrence information in a very large training corpus. Word2vec developed by Mikolov et al. (2013) is one of the popular methods to learn word embeddings. Vectoral representations of words that are semantically similar lies in close proximity in the multidimensional Euclidean space as measured using a distance metric. Similarly, doc2vec algorithm creates dense

vectoral representations of documents. Le and Mikolov (2014) proposed Doc2vec algorithm to represents a document by using the word representations of the document and these representations yields higher classification accuracy than many other document representation methods for document classification tasks (Kim et al., 2019).

One of the unsupervised techniques to capture the semantic similarity between documents is topic modelling. Topic modelling learn latent document representations that can be used to perform text mining tasks such as classification, summarization or information retrieval. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) which is a well-known algorithm for topic modelling, groups similar words together in topics and represents documents as distributions over these topics. The idea behind LDA is to model documents as a distribution of topics where each topic defines a distribution over words. In LDA, the distributions are assumed as Dirichlet distribution which is a continuous generalization of the multinomial distribution.

Performance of document classification systems mainly relies on quality of the document vectors. Therefore, it becomes crucial to create low-dimensional representations of documents that can capture semantic and contextual information about the corpus (Liu et al., 2015).

We present a new approach to document representation by merging two different document representation methods named TopWord2Vec. The first one is the doc2vec representation (Le and Mikolov, 2014) and the second one is the document representation based on the LDA. Our main motivation is to combine the capability of representing each document by word vectors which will incorporate co-occurrence based semantic information between words and the semantic information words, documents and latent topics that characterize the overall corpus. In our method, the latent topics of the corpus are used to enrich the semantics of the document representation. We obtain word distributions per topic and topic distribution per document using the LDA. In ad-

*Corresponding author

✉ halilibrahimcelenli@gmail.com (H.: Çelenli)
ORCID(s): 0000-0001-5369-9677 (H.: Çelenli)

dition to this, all word vectors in each document in the corpus are generated by word2vec algorithm. To create a document representation we multiply LDA based vectors with the word vectors. We experimentally evaluate our approach using the doc2vec representations as the baseline document representation and based on these representations we use well-known classification algorithms such as Support Vector Machines (SVM), k-Nearest Neighbor (KNN), Random Forest (RF), Bernoulli Naive Bayes (BNB) and Logistic Regression (LDA). Experiments show that, by using our proposed document representation approach classification algorithms can achieve higher prediction accuracy values. Furthermore, considering the training times, our approach is faster than doc2vec.

We also investigate the impact of choosing different numbers of topics on the proposed approach. The perplexity which is the measure of the ability of a model to generalize to unseen data (Jiang et al., 2013) is used to adjust the topic numbers to input the LDA. The remaining part of this paper is organized as follows: Section 2 gives related work on the use of word embeddings and topic modeling methods. LDA, word2vec and doc2vec are briefly introduced in section 3. In section 4, our proposed model is described. Experimental setup and results are given in section 5 and section 6. section 7 concludes the paper with a discussion and conclusions.

2. Related Work

Distributed vector representation of words or paragraphs in a continuous space is highly popular in text mining systems lately (Hinton, 1986). One of the inspirational approaches is (Bengio et al., 2003) neural language model which is constituted as a feed forward neural network with a linear projection layer and a non-linear hidden layer. This model learns distributed representations for words. Recurrent Neural Network (RNN) based language model (Mikolov et al., 2010) emphasize the major deficiency of a feed-forward network that has to use limited size of context that needs to be specified before training. To address this, the Recurrent Neural Network Language Model (RNNLM) is presented by Mikolov et al. (2010). This propose a neural network architecture to establish semantically better represented word embeddings by using local and global contexts in learning. Doc2vec approach is proposed by Le and Mikolov (2014) in which they present an unsupervised algorithm that learns fixed-length vector representations from variable length of texts such as paragraphs. They show that the paragraph vector is adequate in capturing the semantics of paragraphs and has ability to overcome many weaknesses of bag-of-words models. Doc2vec model trains a paragraph vector by a linear combination of the word embeddings. (Pennington et al., 2014) proposed GloVe model which captures global corpus statistics directly. To achieve this, they trained a specific weighted least squares model on global word-word co-occurrences.

Although the above mentioned methods have all good performances compare to the traditional bag-of-words models there is still room for improvement. One of the interesting approaches for this is to integrate LDA and word embeddings for a better word or document representation.

Le and Mikolov (2014) propose an unsupervised framework that learns continuous distributed vector representations for different sizes of paragraphs. The paragraph vector can be considered as a memory that remembers what the topic of the paragraph is. This unsupervised method is called as Distributed Memory Model of Paragraph Vectors (PV-DM) due to emphasize that their method can be applied to any phrase, sentence or paragraph. In this method, paragraph vectors are unique among paragraphs. Although each word may have various meaning, each word corresponds to a single word embedding. This can be considered as limitation of the model due to leading to inaccurate semantic representation of paragraph (Wang et al., 2015).

Liu et al. (2015) propose the topical word embedding model which considers each word have different embeddings under different topics. They integrate word-topic relation into word embedding model and so the model discovers the various meanings of a word across different topics. LDA and Skip-Gram Mikolov et al. (2013) models are implemented to obtain word-topic pairs and to learn topical word embeddings respectively. Their study only considers the topic diversity of words.

Ghosh et al. (2016) incorporated topics into a LSTM model, and call the resulting model Contextual LSTM. They concluded that, using topic modelling with vectors, in- stead of supervised topic models, are promising.

Mikolov and Zweig (2012) extracted a topic-conditioned Recurrent neural network language models (RNNLM) by performing LDA to fixed-length block of texts. They provide the topic information directly at the input of the model.

Jiang et al. (2018) applies word2vec to learn word vector. Then they assumed that word vectors with the same topic follow a Gaussian mixture model (GMM) that describe the distributions of all words, in which each Gaussian represents a latent topic. After text generation process, a modified GMM is used to represent texts based on their topics. The weight coefficient of GMM is calculated by the probability that the word belongs to each topic.

Nguyen et al. (2015) show that latent feature representations can be used to improve topic models. The word-topic mapping is improved by incorporating two different Dirichlet multinomial topic models with latent feature vector representations of words trained on very large corpora.

Wang et al. (2016) proposed a hybrid method that combines Word2Vec and LDA to generate not only document-topic relations but also the contextual relationships among words. The performance of their model is investigated by using Support Vector Machines (SVM) on the 20NewsGroups dataset.

3. Preliminaries

This section introduces the relevant models such as LDA and word embeddings that are the components of the proposed system.
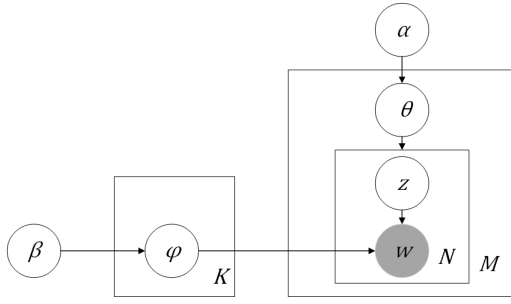
Figure 1: A graphic representation of LDA algorithm.

Figure 2: [a] CBOW and [b] Skip-gram Architectures Mikolov et al. (2013)

3.1. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) (Blei, 2012; Blei et al., 2003) represents a document collection by their thematic content. LDA models a large collection of unlabeled documents as a mixture of topics, where each topic as a distribution over the terms of a fixed vocabulary. The graphical representation of LDA model is depicted in Figure 1.

The only observable variable is represented as w, other represented variables are latent variables. z is denoted as a topic in LDA. The document-topic distribution is denoted as θ , and each one is drawn independently from a symmetric Dirichlet prior α. The topic-word distribution is denoted as ϕ, and each one is drawn from a symmetric Dirichlet prior β. LDA assumes that a document consists K topics, and the generative probability distribution for the documents is

$$p(\phi, \theta, z, w) =$$
$$\prod_{k=1}^{K} p(\phi_k|\beta) \prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N} p(z_{m,n}|\theta_m)(w_{m,n}|z_{m,n}, \phi_k) \quad (1)$$

To obtain the model parameters the posterior distribution is used.

$$p(\phi, \theta, z|w, \alpha, \beta) = \frac{p(\phi, \theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \quad (2)$$

Unfortunately this distribution is intractable to compute due to the marginal probability of the observed data in the denominator cannot be computed exactly. The Collapsed Gibbs sampling is used as an approximate inference technique in our experiments.

3.2. Word2vec

Word2vec (Mikolov et al., 2013) is a shallow neural network which learns low dimensional vectors from trained large text corpus. It predicts words based on their context by using Continuous Bag of Words (CBOW) or Skip-Gram models which are two distinct neural network models. CBOW model predicts the current word based on the context, on the other hand, the skip-gram predicts surrounding words given the current word (Mikolov et al., 2013). The architectures of these models are shown in Figure 2.
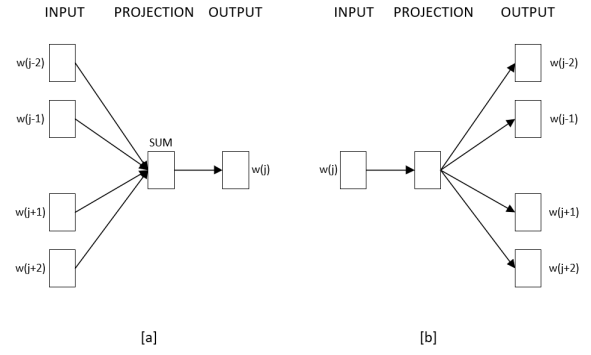
Both neural models learn the underlying relationship between words and their context. By capturing the relationships between words and their context, Word2vec represent words as vectors carrying semantic information.

In this paper, CBOW is used to learn Word vectors. CBOW model predicts a target word w_j using the context words includes previous c words and behind c words of word w_j in a sliding window i.e. $C(w_j) = \{w_j - c, w_j - c + 1, \cdots, w_j - 1, w_j + 1, w_j + 2, \cdots, w_j + c\}$. Given a sequence of training text $D = \{w_1, w_2, \cdots, w_K\}$ The objective, is to maximize the average log likelihood of D.

$$L(D) = \frac{1}{K} \sum_{j=c}^{K-c} \log p(w_j|w_{j-c}, ..., w_{j+c}) \quad (3)$$

For the estimation part the softmax function is used. The softmax is an activation function that produces a probability-based loss value using values from artificial neural networks based deep learning algorithms. The calculation is as follows.

$$log p(w_j|w_{j-c}, ..., w_{t+c}) = \frac{e^y wj}{\sum_i e^y i} \quad (4)$$

Calculations using y(i), which are normalized log likelihood for each output word i, are shown as Equation 5.

$$y = b + U h(w_{j-c}, ..., w_{j+c}; W) \quad (5)$$

In the calculation above, U and b parameters belong to softmax function. The parameter h is formed by the concatenate or averaging of the word vectors.

The purpose of using CBOW instead of skip-gram is that skip-gram calculations are slow and it requires larger amount of the training data to extract context vectors over predicted words. In addition, CBOW architecture has a better performance on the frequently used words (Mikolov et al., 2013) which in our case leads to the topics.

3.3. Doc2vec

Doc2vec, is a model that extracts relations between paragraph vectors. By using again a shollow neural network, the model automatically extracts features that represent the document in a relatively low dimensional Euclidean space. In
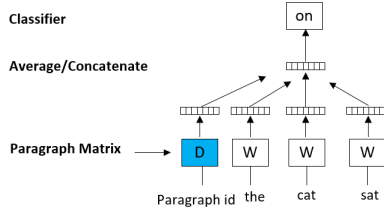
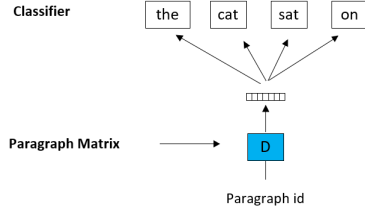**Figure 3:** Estimation of the Word "on" with Distributed Memory Architecture Le and Mikolov (2014)

Figure 4: Estimation of the Word "the","cat","sat","on" with Distributed Bag of Words Architecture Le and Mikolov (2014)

2014, Tomas This distributed representation of documents using dense low dimensional vectors outperform existing bag-of-words representation of documents with high dimensional, highly sparse vectors in several text mining problems (Le and Mikolov, 2014). This model has 2 different architecture which was shown under Figure 3 and Figure 4 named as Distributed Memory (DM) and Distributed Bag of Words (DBOW) architecture.

In this paper, Distributed Memory (DM) is used to learn paragraph vectors. DM architecture was shown on Figure 3 Each paragraph is labeled with a specific paragraph id in the DM architecture. This represents the missing word and increase the success rate of the output. In this approach the correlations are being built between words instead of paragraphs. Here, the D matrix is the representation of the documents, while the W matrix is the unique column matrices used to represent the words. As shown below, the only difference from the y formulation through the CBOW architecture is the addition of the D matrix. Matrix D is a prediction provides based on how often and in what order the words are displayed.

$$y = b + U h(w_{j-c}, ..., w_{j+c}; W, D) \qquad (6)$$

We can consider the architecture of the DM as the implementation of the CBOW architecture on documents. The DBOW architecture is likewise known for its similarity to skip-gram architecture. We have chosen the DM architecture as the doc2vec model because we built the proposed method on CBOW architecture.

The most important property of the paragraph and word vectors is that they are unsupervised meaning that they don't require class labels of the documents. With this method,

no class labeled data is required during the training of the model.

4. Approach

As a novel approach, we incorporate the results of topic modelling with the word embedding model to obtain a document vector which underlies theme of the document. The main steps of our proposed model are as follows:

Step1. The document vectors which include document-topic distributions and topic vectors which include topic-word distributions are obtained by using LDA as depicted in Figure 5.

Topic models are based on the simple idea that given a set of documents $D = \{d_1, d_2, \ldots, d_M\}$ are mixtures of topics $T = \{t_1, t_2, \ldots, t_K\}$, where a topic is a probability distribution over words $W = \{w_1, w_2, \ldots, w_N\}$. Therefore in topic models such as LDA, how well a word represents a topic of a document can be defined with two distributions: topics' distributions under documents which can be denoted as $(T|D)$ and the words' distributions under topics which can be denoted as $P(W|T)$. The probability term $P(T|D)$ measures the likelihood of a topic in a document, and the probability term $P(W|T)$ measures the likelihood of a word in a topic. If a document comes from a specific topic then the likelihood of a topic in that document must be maximum. If a word comes from a specific topic then the likelihood of the word in that topic must be maximum.

For an instance, if the likelihood of a word "student" in a specific topic such as "education" is maximum then the "student" is important for reflecting "education" topic. Besides, if the likelihood of "education" topic in a document which includes "student" is maximum then "student" is important for reflecting "education" topic for that document.

It is assumed that, if a probability distribution of word n under topic k is maximum then the topic of that word is defined as t_k^* Based on this topic information, a new weight degree which denoted as a_n^m and defined as in Equation 7 is proposed to represent the importance of word n in document m.

$$t_{k*} = \arg\max P(w_n \mid t_k); k = 1, \ldots, K \qquad (7)$$

$$a_n^m = P(w_n \mid t_{k*}) \times P(t_{k*} \mid d_m); n = 1, ..., N$$

The higher the value of α_n^m, the more decisive the w_n for the document m is.

Step2. The word embedding vectors are obtained by using word2vec. The learned word vectors underlie relationship between words and their context. This information considered as a semantic knowledge about the words in the corpus. Word2vec vectorizes each word in corpus into a fixed length vector. After word2vec training, words in vocabulary is represented as $V = \{v_1, v_2, ..., v_N\}$.

Step3. The semantic representation of the words is strengthened by combining the results of the previous two steps. The numerical values of the word embedding vector obtained in the second step and the probability distributions obtained in
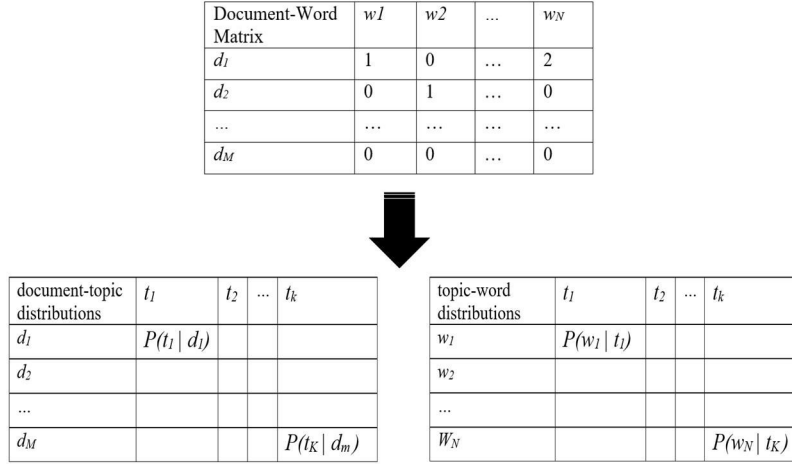
Document-Word Matrix	w1	w2	...	wN
d1	1	0	...	2
d2	0	1	...	0
...	...	...	...	...
dM	0	0	...	0

document-topic distributions	t1	t2	...	tk
d1	$P(t_1 \mid d_1)$			
d2				
...				
dM			$P(t_K \mid d_m)$	

topic-word distributions	t1	t2	...	tk
w1	$P(w_1 \mid t_1)$			
w2				
...				
WN			$P(w_N \mid t_K)$	

Figure 5: Outputs of LDA: Document Vectors (document-topic distributions) and Topic Vectors (topic-word distributions)

the first step are multiplied to modify the word embedding vector. Lets $V' = \{v'_1, v'_2, ..., v'_N\}$ represents the modified word embedding vector for document m. Modified weight of word n in document m, is obtained as in Equation 8.

$$v_{n^I} = v_n \times \alpha_{n^m}; n = 1, \dots, N \qquad (8)$$

Step4. At last, the document vector is created by combining modified word vectors. Document vectors generated by summing word vectors and divide them to the numbers of words in documents afterwards as in Equation 9.

$$d_m{}^I = \frac{\sum_{v_n^I} \in d_m^{v_n^I}}{\mid d_m \mid}; m = 1, \dots, M \qquad (9)$$

These steps are applied to all documents in the corpus and the document vectors are obtained by our proposed approach. After the completion of these steps, document vectors and their associated class labels are used to train traditional machine learning classifiers such as SVM, KNN, Random Forest, Bernoulli Bayes and Logistic regression classifiers respectively. The proposed model is illustrated in Figure 6.

5. Experimental Design

5.1. Data Sets and Preprocessing

We use several Turkish and English text classification datasets to evaluate our approach. The Turkish Datasets used in this study are; 1150News, and 3000Tweet which are provided by Yildiz Technical University Kemik NLP Research Group[1]. We also used TTC-3600 (Kılınç et al., 2017) and 9127Tweet[2] datasets. The datasets consist of tweets such as 3000Tweet and 9127Tweet includes 12 unique words per document on average, while news documents includes 228 words per document , thus they can be considered as short text and long text datasets, respectively.

10 million tweets are used together with four of Turkish datasets with an overall number of 10.073.869. Table 1 summarizes the Turkish datasets used in experiments.

Comparing to the Turkish datasets, English datasets are only consists of tweet data. Archeage, Iphone6, Hobbit as Sent Collection Datasets and Stsgold Saif et al. (2013) datasets[3] are used. There are also unlabeled tweets for training purposes. All the English datasets consists of 1.882.876 data. Table 2 summarizes these datasets.

The process starts with the preprocessing steps. During preprocessing, punctuation, numerical characters and stopwords are removed. Stemmer is not used. After the completion of data preprocessing, the tokenized words in test data and corpus data are sent to LDA and Word2vec respectively. Finally, by combining LDA with Word2vec model, document vectors are created. The applied preprocessing steps are shown in Figure 7.

As indicated in Figure 7, the corpus and test data was used separately during training. For instance, if the test data is 1150 news dataset then the corpus data is constituted with 1150 news + 3000 tweets + 9127 tweets + Ttc 3600 + 10 million tweets datasets. The corpus data must include the test data due to capture topic words by word embedding methods.

5.2. Parameter Selection

The parameter selection is important for modelling an optimum hypothesis in both machine learning and deep learning. This section the optimum parameter settings are summarized for the algorithms which are used in our experiments.

Word2vec and Doc2vec models use CBOW and DM models respectively, with the parameters: vector size is 100, window length is 5, min count is 1, training epochs are 5 and negative samples are 5.

The parameters used in LDA are: topic number is set to 10; the number of passes is set to 1000; the parameter
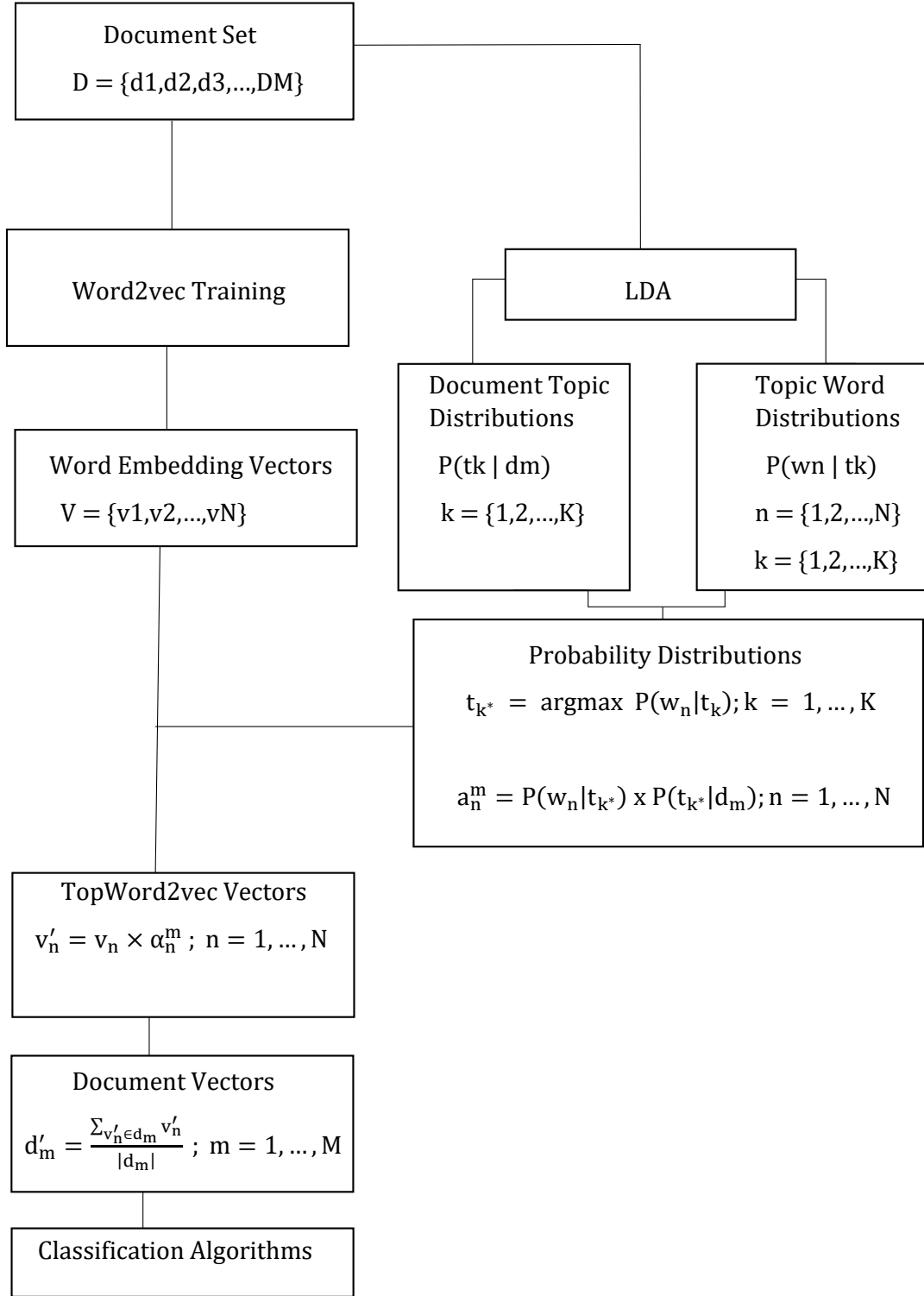
---

[1] www.kemik.yildiz.edu.tr/?id=28
[2] www.kemik.yildiz.edu.tr/?id=28

[3] http://lasid.sor.ufscar.br/ml-tools/

Figure 6: The Proposed TopWord2vec Model

Table 1
Properties of the Turkish datasss

Dataset	Description	Number of Classes	Vocabulary Size (Number of Words)	Mean Length (Number of Words)
1150News	Consists of 1150 news data	5	50.973	218
3000Tweet	Consists of 3.000 tweets data	3	12.040	10
9127Tweet	Consists of 9.127 tweets data	2	26.682	16
Ttc-3600	Consists of 3.600 news data	6	104.774	239

Table 2
Properties of the English datasetss

Dataset	Positive	Negative	Sum Data	Vocabulary Size (Number of Words)	Mean Length (Number of Words)
Archeage	724	994	1.718	2.945	13
Iphone6	371	161	532	1.439	10
Hobbit	354	168	532	1.311	13
Sts-gold	632	1.402	2.034	4.609	12

*Corpus = Test + Train
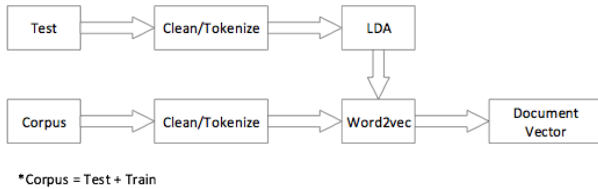
Figure 7: Word2vec+LDA General Representation

Table 4
Perplexity of LDA with different numbers of topics for English datasets

	10 Topics	20 Topics	50 Topics	100 Topics
Archeage	-38.42	-46.34	-61.64	-73.85
Iphone6	-65.46	-73.34	-100.65	-101.03
Hobbit	-48.49	-58.44	-76.16	-85.92
Sts-gold	-50.51	-61.42	-79.10	-90.83

α is set to 50/number of topics; the parameter β is set to 0.01. The perplexity (Blei et al., 2003) which measures the ability of a LDA model to generalize to unseen data is used to investigate the impact of different numbers of topics on the proposed approach. It is observed that, our method achieves the best overall performance when the topic number is 10 for both Turkish and English datasets as it is seen in Table 3 and Table 4 respectively.

Table 3
Perplexity of LDA with different numbers of topics for Turkish datasets

	10 Topics	20 Topics	50 Topics	100 Topics
1150News	-48.56	-55.67	-73.77	-64.83
3000Tweet	-64.62	-70.98	-77.56	-80.52
9127Tweet	-35.95	-40.14	-41.76	-42.84
Ttc-3600	-30.87	-35.52	-40.28	-36.11

6. Experimental Results

In our experiments, LDA is used to extract latent topics and word2vec is used to extract word vectors simultaneously.

For both LDA and Word2vec we use Gensim implementations (Rehurek and Sojka, 2010).

The proposed method is applied to create Topic Words Vectors, which we name TopWord2vec, of the documents in the datasets. After that, the latent document representations which is learnt by TopWord2vec approach is used in classification experiments with well-known supervised learning algorithms such as Linear SVM, KNN (k=3), Random Forest, Bernoulli Bayes and Logistic Regression.

The classification results are presented using accuracy as our main evaluation metric. This is due to the fact that F-measure and accuracy values from the experiments are very close. Table 5 shows the classification results of SVM algorithm for Turkish news and tweet datasets.

When Table 5 is examined, it can be seen that the results obtained by using TopWord2vec vector in SVM is better than the results obtained by using Doc2vec in SVM for all Turkish datasets.

Table 6 shows the results of KNN algorithm, and the results indicate that, KNN algorithm provided a minimum increase of 6% in Turkish datasets by using proposed TopWord2vec instead of Doc2vec.

Table 5
Classification accuracies with SVM

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
1150News	90.26	88.43
3000Tweet	58.79	47.60
9127Tweet	79.84	74.82
Ttc-3600	87.39	86.50

Table 6
Classification accuracies with KNN

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
1150News	88.52	82.96
3000Tweet	49.36	33.80
9127Tweet	77.45	64.91
Ttc-3600	84.56	78.42

The classification results of Random Forrest algorithm are shown in Table 7. The proposed method gives better classification results than doc2vec model. The Random Forrest algorithm provided a minimum increase of 6% by using TopWord2vec in Turkish datasets.

Table 7
Classification accuracies with Random Forest

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
1150News	85.04	77.22
3000Tweet	49.23	41.29
9127Tweet	76.21	70.06
Ttc-3600	82.61	74.19

Table 8 shows linear Bernoulli Bayes algorithm classification results by using Turkish news and tweet data. As in other classification algorithms, TopWord2vec gives better results than Doc2vec. There is no increase in the datasets of 1150News and Ttc-3600 on the Bernoulli Bayes algorithm, but an increase of 3000Tweet and 9127Tweet data has been observed.

Table 8
Classification accuracies with Bernoulli Naive Bayes

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
1150News	87.48	87.13
3000Tweet	53.91	40.43
9127Tweet	76.33	64.48
Ttc-3600	81.31	80.89

Table 9 shows the classification results of linear Logistic Regression algorithm by using Turkish news and tweet data.

There is no increase in the Ttc-3600 dataset on the Logistic Regression algorithm, but an increase in other datasets has been observed.

Table 9
Classification accuracies with Logistic Regression

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
1150News	89.13	84.09
3000Tweet	58.99	47.26
9127Tweet	79.33	75.41
Ttc-3600	86.50	86.28

In general, when the classification results for Turkish datasets are examined, it is seen that, the SVM algorithm produces the superior results when it is used with TopWord2vec instead of Doc2vec

Table 10 shows the linear SVM algorithm classification results by using English tweet data. While there is no increase in Archeage and Sts-gold data, there is a decrease in Iphone6 and Hobbit datasets.

Table 10
Classification accuracies with SVM

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
Archeage	77.34	75.15
Iphone6	71.75	73.11
Hobbit	77.12	73.73
Sts-gold	79.12	78.17

Table 11 shows the linear KNN algorithm classification results by using English tweet data. TopWord2vec gives better results than Doc2vec. In particular, there was an 9% increase in Archeage dataset.

Table 11
Classification accuracies with KNN

| Datasets | Classification Accuracies (%) | |
	TopWord2vec	Doc2vec
Archeage	74.80	66.06
Iphone6	73.08	71.82
Hobbit	75.05	70.08
Sts-gold	73.76	69.32

Table 12 shows the linear Random Forest algorithm classification results by using English tweet data. TopWord2vec gives better results than Doc2vec. In particular, there was an 11% increase in Hobbit dataset.

Table 13 shows the linear Bernoulli Bayes algorithm classification results by using English tweet data. TopWord2vec gives better results than Doc2vec. In particular, there was a 12% increase in Iphone6 dataset.

Table 12
Classification accuracies with Random Forest

Datasets	Classification Accuracies (%)	
	TopWord2vec	Doc2vec
Archeage	72.42	68.05
Iphone6	70.43	66.16
Hobbit	76.22	65.13
Sts-gold	73.11	69.47

Table 13
Classification accuracies with Bernoulli Naive Bayes

Datasets	Classification Accuracies (%)	
	TopWord2vec	Doc2vec
Archeage	71.00	63.88
Iphone6	70.64	61.28
Hobbit	72.04	62.04
Sts-gold	73.41	69.47

Table 14 shows the linear Logistic Regression algorithm classification results by using English tweet data. TopWord2vec gives better results than Doc2vec. There was no increase in the Iphone6 dataset, but an increase in other datasets was observed.

Table 14
Classification accuracies with Logistic Regression

Datasets	Classification Accuracies (%)	
	TopWord2vec	Doc2vec
Archeage	77.90	75.44
Iphone6	71.18	70.66
Hobbit	78.50	71.24
Sts-gold	79.40	77.29

In general, document representations, e.g. document vectors obtained by our proposed approach TopWord2vec yields better classification results by a wide range of classification algorithms compare to the doc2vec representations on a large number and variety of text classification datasets in two different languages; English and Turkish.

Yet another benefit of our proposed approach is better computational time complexity. When we measure the training times (Table 15) we see that it is faster to train Top-Word2vec model.

Table 15
Training time in minutes

	TopWord2vec	Doc2vec
Turkish Dataset	25 m.	367 m.
English Dataset	3 m.	14 m.

7. Conclusion

In this paper, we propose a new document vector representation based on topic and word distributions of a given document combined with word embedding vectors of the words of this document. LDA and Word2Vec methods are integrated for the proposed document vector which is called as TopWord2vec. The proposed algorithm to create document vectors not only make use of co-occurrence relations between words, but also considers the theme of the document and words which are good representatives of these themes. The experimental results show that document representations, e.g. document vectors obtained by our proposed approach TopWord2vec yields better classification results by a wide range of classification algorithms on a large number and variety of text classification datasets in two different languages; English and Turkish, compare to the doc2vec representations.

As a future plan, we aim to analyze whether the Top-Word2vec representation improves the performance of information retrieval from the documents. Additionally, we also aim to implement TopWord2vec document vectors in unsupervised machine learning models.

8. Acknowledgement

References

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A neural probabilistic language model. Journal of machine learning research 3, 1137–1155.

Blei, D.M., 2012. Probabilistic topic models. Communications of the ACM 55, 77–84.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. Journal of machine Learning research 3, 993–1022.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., Heck, L., 2016. Contextual lstm (clstm) models for large scale nlp tasks. arXiv preprint arXiv:1602.06291 .

Hinton, G., 1986. Distributed representations//parallel distributed processing: Explorations in the microstructure of cognition. volume i. foundations/ed. by de rumelhart & jl mcclelland–cambridge.

Jiang, B., Li, Z., Chen, H., Cohn, A.G., 2018. Latent topic text representation learning on statistical manifolds. IEEE transactions on neural networks and learning systems 29, 5643–5654.

Jiang, D., Leung, K.W.T., Ng, W., Li, H., 2013. Beyond click graph: Topic modeling for search engine query log analysis, in: International Conference on Database Systems for Advanced Applications, Springer. pp. 209–223.

Kılınç, D., Özçift, A., Bozyigit, F., Yıldırım, P., Yücalar, F., Borandag, E., 2017. Ttc-3600: A new benchmark dataset for turkish text categorization. Journal of Information Science 43, 174–185.

Kim, D., Seo, D., Cho, S., Kang, P., 2019. Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. Information Sciences 477, 15 – 29. URL: http://www.sciencedirect.com/science/article/pii/S0020025518308028, doi:https://doi.org/10.1016/j.ins.2018.10.006.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: International conference on machine learning, pp. 1188–1196.

Liu, Y., Liu, Z., Chua, T.S., Sun, M., 2015. Topical word embeddings, in: Twenty-Ninth AAAI Conference on Artificial Intelligence.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S., 2010. Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association.

Mikolov, T., Zweig, G., 2012. Context dependent recurrent neural network language model, in: 2012 IEEE Spoken Language Technology Workshop (SLT), IEEE. pp. 234–239.

Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M., 2015. Improving topic models with latent feature word representations. Transactions of the Association for Computational Linguistics 3, 299–313.

Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

Rehurek, R., Sojka, P., 2010. Software framework for topic modelling with large corpora, in: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Citeseer.

Saif, H., Fernandez, M., He, Y., Alani, H., 2013. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold .

Wang, Q., Liu, R., Li, H., Guo, W., 2015. Topical paragraph vector learning, in: 2015 11th International Conference on Natural Computation (ICNC), IEEE. pp. 182–187.

Wang, Z., Ma, L., Zhang, Y., 2016. A hybrid document feature extraction method using latent dirichlet allocation and word2vec, in: 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), IEEE. pp. 98–103.

LaTeX Source Files

*Declaration of Interest Statement

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: