Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 22, Sayı 2, 569-576, 2018

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 22, Issue 2, 569-576, 2018

DOI: 10.19113/sdufbed.15893

# A Comparison of Different Approaches to Document Representation in Turkish Language

**Savaş YILDIRIM**[*1], **Tuğba YILDIZ**[1]

[1]İstanbul Bilgi Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, 34060, İstanbul

(Alınış / Received: 28.12.2017, Kabul / Accepted: 04.06.2018, Online Yayınlanma / Published Online: 03.07.2018)

**Keywords**
Document representation,
Deep learning,
Natural language processing

**Abstract:** Recently, deep learning methods have demonstrated state-of-the-art performance in numerous complex Natural Language Processing (NLP) problems. Easy accessibility of high-performance computing resources and open-source libraries makes Artificial Intelligence (AI) approaches more applicable for researchers. This sudden growth of available techniques shaped and improved standards in the field of NLP. Thus, we find an opportunity to compare different approaches to document representation, owing to various open-source libraries and a large amount of research. We evaluate four different paradigms to represent documents: Traditional bag-of-words approaches, topic modeling, embedding based approach and deep learning. As the main contribution of this article, we aim at evaluating all these representation approaches with suitable machine learning algorithms for document categorization problem in the Turkish language. The supervised architecture uses a benchmark dataset specifically prepared for this language. Within the architecture, we evaluate the representation approaches with corresponding machine learning algorithms such as Support Vector Machine (SVM), multi-nominal Naive Bayes Algorithm (m-NB) and so forth. We conduct a variety of experiments and present successful results for the Turkish document categorization. We also observed that tradition approaches have still comparable results with Neural Network models in terms of document classification.

# Metin Temsil Yöntemlerine Yönelik Farklı Yaklaşımların Karşılaştırılması

**Anahtar Kelimeler**
Metin temsiliyeti,
Derin öğrenme,
Doğal dil işleme

**Özet:** Son zamanlarda derin öğrenme mimarileri bir çok doğal dil işleme problemini başarılı bir şekilde çözmüştür. Açık kaynak kodlu kütüphanelerin yaygınlığı yapay zeka yaklaşımlarını daha uygulanabilir hale getirmiştir. Teknolojideki bu ani ivmelenme doğal dil işlemedeki standartları dönüştürdü ve geliştirdi. Bu çalışmada açık kaynak kodların ve alanla ilgili araştırmaların rahat erişebilirliği sayesinde metin temsiliyeti yaklaşımlarının önemli bir kısmını değerlendirme imkanı bulduk. Dört farklı paradigmayı metin temsiliyeti açısından değerlendirdik: Geleneksel kelime torbası yaklaşımı, konu modelleme, gömme temsiliyeti ve derin öğrenme. Çalışmanın ana katkısı olarak, Türkçe için metin sınıflandırma problemini tüm bu metin temsiliyetlerini ve ilgili makine öğrenme algoritmalarını kullanarak ele aldık. Oluşturulan denetimli öğrenme mimarisi özellikle Türkçe için hazırlanmış bir veri seti ile sınanmıştır. Her bir temsiliyet için onunla uyumlu çalışacak SVM, çok-katlı Naive Bayes (mNB) gibi makine öğrenmesi algoritmaları sınandı. Çeşitli deneyler sonucunda başarılı bir metin sınıflandırıcı mimarisinin Türkçe için nasıl kurulacağını bu makalede tartıştık ve başarılı modeller sunduk. Son olarak kelime torbası gibi geleneksel yöntemlerin hala başarılı olduğunu ve derin öğrenme temelli modellerin bazılarından daha iyi olduğunu gördük.

## 1. Introduction

Words are needed to be represent in vector space models (VSM) for the Natural Language Processing (NLP) problems. For years, VSM has been used in the field of NLP to compute semantic similarity. First, [24] represented a word as a real-valued vector by using co-occurrence statistics to measure the semantic similarity. It is based on the idea that if two documents or words share similar neighboring words, they are considered similar. The similarity between the vectors of the words are simply computed by cosine similarity and other metrics. The approach uses a fixed-length representation of a document within document-term matrix. It is also called the bag-of-words (BoW) where the bag contains the words of a document by ignoring word order in it.

The main drawback of traditional BoW approach is high

* *Corresponding author: savas.yildirim@bilgi.edu.tr*

dimensionality. A variety of approaches have been applied to dimensionality reduction. The common way is the feature selection by exploiting some selectional criteria. Some other kind of paradigms namely topic modeling have been also used for reduction. Latent Semantic Indexing (LSI) (or Latent Semantic Analysis-LSA) have been mostly applied to such document-term matrix to reduce the dimension [9]. Another popular and widely used topic modeling is Latent Dirichlet Analysis (LDA) that is a new variant LSI-based paradigm especially for textual data [7]. It can be considered another document representation model where a stochastic gradient optimization algorithm clusters the documents based on word co-occurrence statistics and this makes a new representation for textual data. There is Probabilistic Latent Semantic Analysis (PLSA) that employs probabilistic method rather than using matrices.

Recently, neural network language models (NNLM) have demonstrated promising performance by reducing time complexity and successfully solved many NLP problems [18]. They effectively generate dense and short embeddings, namely word embeddings [22, 23]. For document embeddings, averaging word embeddings in a document could be a way for the representation. On the other hand, [17] proposed another method to produce document vector which is similar to word embeddings. The method directly produces document embeddings along with the word embeddings. They found document embedding very effective for the problem of sentiment analysis and document retrieval [17]. There are some deep learning architectures such as the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN) [19]. They implicitly map the documents layer by layer and classify the them.

The motivation is to evaluate these representation model for the problem of document classification in Turkish language. The study can be accounted the first attempt that compares these approaches for Turkish document classification problem. We designate a supervised architecture by using a benchmark dataset in which the documents are collected from news corpus and are labeled with categories. The different representation techniques from NNLM to traditional approaches have been applied to the documents. Then, the architectures trained the model using these new representations as predictors and the corresponding categories as target class. We measured how effectively each paradigm represents the documents for the problem. We report our results with a detailed analysis as described in the following section.

## 2. Turkish Document Classification in Literature

There exist a variety of studies for the Turkish document classification problem. These studies have examined the topic, authorship and gender detection. As one of the earliest studies, [4] handled document classification problem using n-gram and BoW approach. In order to find authorship, specify the genre and gender, four different machine learning algorithms have been applied and been received successful results. This study can be considered as one for the first attempt for the Turkish language. As one another earliest study, [29] applied classification methods for 18 different authors, 35 different document for each author. They compared 5 different classification algorithms using n-gram and BoW approach and addressed that using n-gram with other features highly improves the system performance. They found the Artificial Neural Network (ANN) and Support Vector Machine (SVM) better than other algorithms. The most comprehensive study examining the traditional methods has been done by [3]. They evaluated the contribution of the document representation techniques through six different text classification problems. The problems are detection of four different moods from a given text, three different sentiments from movie reviews, detection of authorship out of 18 journalists, gender identification, detection of 5 different news categories and detection of poet of a given poem out of 4 poets.

Recently [16] constructed a benchmark dataset, namely TTC-3600, including 6 different classes in order to study document classification problem. They used BoW, n-gram approaches and feature selection method with the dataset. Six different machine learning algorithms have been evaluated. Stemming and Attribute ranking-based Feature Selection (ARFS) has been utilized to improve the performance. They concluded that the Random Forest (RF) algorithm with Stemming and ARFS has performed better than other configuration. Some studies especially examined the preprocessing phase to evaluate its contribution to the models. [27] measured the contribution of using longest and smallest stem suggested by Zemberek [1] library to the classifiers performance where the classifiers are RF and NB. They concluded the former is better then the latter. Another study measured the contribution of stemming to the model performances [2]. However, the number of examples and the classed are not sufficient level. They noted that the they did not find statistically significant contribution of using stemming. On the other hand, they alternatively applied very simple and different stemming approach which takes only the first K characters of a given word, FPS (Fixed Prefix Stemming). They interestingly noted that the FPS outperformed the traditional stemming where the optimum K is found 5 where the machine learning algorithms are SVM and NB. Another important study examining the contribution of preprocessing phase has been conducted by the study [28]. Stemming, stopword elimination and feature weighting were among the techniques applied to Turkish text classification. They applied Zemberek and FPS7 stemming approaches and concluded that the stemming has been found very useful for the information retrieval but the text classification. They did not see any contribution to the problem.

[30] examines the preprocessing phase and its contribution through the domain, natural language and dimensionality

[1] github.com/ahmetaa/zemberek-nlp

reduction. Two different languages and different domain information has been used within a well designed architecture. They concluded that a improper system designed might lead to underestimation and poor results as well. [31] examining stemming phase has used information gain and Naive Bayes (NB) classifier. They found that the contribution of stemming to the text classification was quite limited.

Other than the traditional approaches, there exist some alternative approaches designed for the Turkish language. [1] has utilized Hidden Dirichlet Analysis (HDA), LSA and LDA and discussed the effect of dimensionality reduction for the problem. They compiled an annotated corpus including academic articles written in Turkish. The system has been tested for two different datasets in which there are 18 and 34 classes respectively. Although they found stemming useful using the classifiers SVM, NB and RF, however we observed that the results obtained are very low against other studies in Turkish language. As a different study, [10] utilized word semantic analysis for the text classification problem. The study constructed a new representation model using semantic similarity between words and solved text classification problem within the semantic space. The words have been clustered using euclidean distance and the cluster membership information has been used to create new dimensions for the documents. Consequently, the documents are then represented in a smaller dimension. With the dimension size of 100, they got 92.5 % success rate using Logistic Regression (LR) classifier.

Ensemble learning is considered an important learning approach in the field of machine learning, even though it suffers from over-fitting in some cases. [5] compared the ensemble learning with traditional machine learning approaches using two different a 6-classes datasets for the text classification problem. In order to reduce running time complexity they utilized pruning techniques. They pruned the forest with the rate of 90% even without losing success.

## 3. Document Representation and The Methodology

Supervised learning models require tabular data format in which it induces a function between the independent variables and the dependent variable. Thus, the documents are first needed to represent in a fixed sized tabular data along with their categories such as economy, sport so forth. There are variety of ways to represent the document in a fixed-sized vector. We applied all these techniques as listed below to our document classification problem in Turkish language.

### 3.1. BoW model

Vector space models are used to represent a document or a word by embedding it into a vector space. For years, VSM has been used in the field of NLP to compute semantic similarity. First, [24] represented a word as a one-hot vector by counting co-occurrence statistics with other words where the dimension of the matrix is equal to the vocabulary used. Fixing the size of the vector makes easier to compute the similarity and other metrics. To compare two documents or words, the cosine similarity function is applied to the vectors of same size. This fixed-sized approach creates document-term matrix where rows indicate documents and columns show words. It is also called the bag-of-words (BoW) where the bag contains the words of a document by ignoring word order in it.

The main disadvantage of BoW representation is the enormous number of terms that is equal to the size of the vocabulary. As the dimension of the vector exceedingly increases, so does computational complexity of the designed system. The widely applied solution is the feature elimination in the preparation step. It discards non-informative terms based on some metrics using corpus statistics. [25] pointed that the frequent terms could be informative. Some selectional criteria such as chi-square ($\chi^2$) are found very effective to find informative terms from corpus, [20, 21, 25].

Some studies addressed that the most effective selectional criteria is Information Gain (IG) [21, 25]. It measures how many number of bits of information the presence or absence of a word in a document contribute to model accuracy. $\chi^2$ is another widely used formula in many field such as statistics. It tests the lack of independence between a word and a category using document-term table. Some other selectional criteria such as point-wise mutual information (PMI) and DICE are also used.

### 3.2. Topic modeling

As one of the dimensionality reduction approaches, Latent Semantic Indexing (or Latent Semantic Analysis) have been widely applied to document-term matrix to reduce the dimension and produce informative and short latent dimension. LSI uses Singular Value Decomposition (SVD) as a method for building significant dimensions derived from a document-term matrix [9]. It is a member of a method family that can approximate an N-dimensional matrix using fewer dimensions, including Principle Components Analysis (PCA), Factor Analysis etc, [14, 15, 26]. Some latent indexing approaches do not use matrices. For instance, PLSA employs probabilistic method rather than using matrices. It is also called probabilistic latent semantic indexing.

LDA can be considered another document representation model where the stochastic algorithm clusters the documents based on co-occurrence statistics [7]. It represents documents as a list of discovered topics. Topics and their probabilities are learned as discrete distributions where the topics consist of a set of words. The models takes the topic size and words size as parameter before training phase. The documents are then represented by means of latent semantic structures, topics. Contrary to feature selection models, LSI and LDA does not explicitly use the words as dimension, but create most informative latent dimensions by word composition instead.

### 3.3. Document and word embeddings

Recently, NNLM have gained big attention and demonstrated promising performance by reducing time complexity. The most important characteristics of NNLM is its capacity of generating dense and short embeddings, namely word embeddings [22, 23]. In the neural networks architecture, each word is initially associated with a random vector. As a two-layer neural network processes textual corpus, the vectors are iteratively updated by applying stochastic gradient descent (SGD) where the gradient is measured by back-propagation. The objective is to guess the last word from a given word sequence. Thus, the prediction task is typically similar to multi-class classification where soft-max function is used to compute class probability estimation. The network finally learns the embeddings for all words appeared in the corpus by convergence.

As one of the most popular word embeddings models, word2vec model showed how word embeddings were efficiently trained within two different architectures, namely Continuous Bag of Words (CBoW) and the Skip-gram (SG) [22]. The architecture achieved both minimizing computational time complexity and maximizing model accuracy. As second model, [23] proposed another word embedding model, namely glove. It is based on matrix factorization and a new global log-bilinear regression model that combines the advantages of the two important models in the literature: global matrix factorization and local context window methods. These two popular word embedding models also proved that embeddings are very good at capturing syntactic and semantic regularities, using the vector offsets between word pairs.

Naturally word embeddings also help to improve document representation. Averaging all word embeddings in a document is considered possible representation of the document. On the other hand, [17] presented another neural network based approach to train document embeddings, called paragraph vector. Learning paragraph vector is highly inspired by the neural network architecture of word embeddings, word2vec. The architecture trains the vectors by a process that predicts the last word using other words in a given context. The network uses a fixed-length context by a sliding window with a size of K. The paragraph vectors are learned in a similar manner where each paragraph is initially associated with a random vector and added to head position of each contextual window. And the architecture tries to predict the last word using all vectors of the words in the context plus the paragraph vector. The architectures use either averaging or concatenation of the vectors.

The paragraph token is shared across all contexts generated from the regarding paragraph but not across the paragraphs. However, the word vectors are shared across paragraphs. As the paragraph token acts as a memory and its vector is always added to each context, this model is called the Distributed Memory Model of Paragraph Vectors (PV-DM). The paragraph vectors and word vectors are trained using SGD and the gradient is obtained via back-propagation. At every step of SGD, the error gradient is computed via NNLM and the parameters of the model is updated using the gradient. An alternative way proposed is to ignore the window contextual words and rather randomly selects a fixed-number of words from the paragraph in hand. At each iteration of SGD, a random word is selected as target class, remaining sampled words and the paragraph token are used as predictors as in the multi-class classification task. Since that it is very similar to bag-of-word approach, it is called the Distributed Bag of Words version of Paragraph Vector (PV-DBoW). Moreover it can be considered the counterpart of skip-gram (SG) model of word2vec implementation.

### 3.4. Deep learning

Word embedding methods learn a real-valued vector representation in a fixed sized vocabulary. Word2Vec or Glove is a two-layer shallow neural network that takes textual corpus and produces a set of word vectors. Therefore Word2vec is not considered a good example of deep learning. But such methods can turn documents into a real-valued vectors that deep learning networks can understand. The CNN and RNN which are the two main types of deep neural network architectures, are utilized for addressing to various NLP tasks [19]. The CNN is a feed-forward network equipped with convolution layers interleaved with pooling layers. Max pooling is used to reduce the number of parameters within the model and generalizes the results from a convolutional filter. The RNN is used for modeling units in sequence and temporal dependencies [12]. While CNNs are mentioned as hierarchical architecture, RNNs are sequential architectures [32]. However, simple RNNs have difficulties to capture long-term dependencies because of vanishing or exploding gradient [6, 13]. One of the solutions resolved the vanishing and exploding gradient problem using gating mechanisms which have been developed to alleviate some limitations of the standard RNN, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) [13], [8]. We applied all these deep learning algorithms using python keras libraries with the default parameter.

## 4. Experimental Setup

### 4.1. Data

In this study, we used a benchmark dataset compiled from Turkish newswire documents under seven different categories; world, economy, culture-art, health, politics, sports and technology. There exist 700 documents under each category and 4900 documents in total. The total number of tokens in the corpus is over 1.3 M. The data was obtained from publicly available web site [2]. The white space characters, all digits, punctuations were removed. Some collocations are obtained and annotated such as Istanbul_Belediyesi. The collocation phrases are captured by log-likelihood ratio of the probability. We only used surface form of the terms and the number of unique terms are about 100 thousands.

[2] www.kemik.yildiz.edu.tr

### 4.2. Representations

Contrary to other paradigms, feature selection based BoW approaches require three separated data; development set, training set and test set. We selected one third of documents as development set. Remaining data is divided by 10-fold cross validation. The first development set is exploited to specify informative words to reduce dimensionality. Feature selection metrics use this first set to rank and eliminate the words. We observed and report that IG and $\chi^2$ are the most powerful selectional criteria. And the all documents are represented by those selected informative words. The second set is used to train classifiers and the last test set is used to measure model performance. It is typically similar to the validation phase of machine learning, such as K-fold cross validation.

Topic modeling simply represents the documents in a dimension of given size. Both LSI and LDA require the size of dimension that is the number of topics to train the model. We set the dimension size to 200 and 400 respectively. Each dimension is actually probabilistic composition of the most contributing words. We set the size of the words to 10. Topic modeling produces numerical variables for the representation in the end.

Word embeddings averaging is a possible way of the representation. We exploited two important embeddings models for word embeddings; word2vec and glove where dimension size is set to 300. The documents are represented by averaging the vectors of words. There is also alternative and more effective way of document-specific embeddings, namely doc2vec. We prepared doc2vec document embeddings architecture that consists of two different settings: Distributed Memory Model of Paragraph Vectors (PV-DM) and Distributed Bag of Words version of Paragraph Vector (PV-DBoW). These two models suggest different document representation. Concatenation of these two representation can also be used as a third representation. Finally, we applied three representations of document embeddings: PV-DM, PV-BoW and PV-DM+BoW. Embedding based models, word embeddings averaging and document embeddings, also produce numerical variables for the representation.

### 4.3. ML algorithms

Especially for BoW representation, mNB has found an apparently effective classification method for text classification so far. It has been found competitive with the state-of-the-art algorithms in many studies [21]. We applied mNB by setting smoothing parameter alpha to 1, by default. Since mNB requires non-negative continous variable, we can only apply it to BoW representation but other representation model since that other representation might produce negative value in the vectors. SVM has recently been found another popular machine learning algorithm for many data mining problems. Many researchers also proved that performance of linear kernel of SVM is better than that of radial and Gaussian especially for Text Categorization problem. It is quite similar to perceptron. While the perceptron is used to minimize misclassification errors, the objective of SVM is to maximize the margin which is the distance between the separating planes. The objective function maximizes the margin between the decision boundaries. When SVM is used with linear separator, it is called linear SVM. There exist non-linear alternative for the SVM, called kernel SVM. We used SVM with linear kernel where the regularization is L2, penalty parameter C is set to 1 and loss function is squared-hinge. These are mostly default parameter of python sklearn library. Another appropriate algorithm is Logistic Regression algorithm. This algorithm derived from linear regression by the application of sigmoid function. This algorithm is found very useful since that it predict class estimation probability as well. While it applies linear separation, There is also non-linear variant for the logistic regression. We chose L2 penalization, set C parameter to 1 for logistic regression algorithm.

Decision Tree (DT) is another widely used machine learning algorithm because of its interpretability. It applies divide-and-conquer mechanisms to derive a decision tree. It learns a list of question using features as question. In an iterative process, the data is recursively divided into two or more until the leaves are pure. It means that it creates new node until the examples in divided subset belong to same class. The features are needed to sort depending on their importance where there exists two popular impurity measure gini and information gain. We select gini as the impurity measure. The minimum number of sample for a leaf is set to 2. The last algorithm is selected from lazy learner, KNN. The algorithm does not induce a model. Instead, during the decision time, it compares a given instance to the remaining dataset. Therefore it does not have training time but has big testing time. The algorithm take the first K neighbors and decide the class based on majority class of the neighbors. For our experiment, we set K to 3 and select distance function as euclidean.

### 4.4. Implementations

All the implementations based on four different paradigms discussed above were applied to same benchmark Turkish dataset in which there are total 4900 documents labeled with seven different categories. The experiments for word embeddings, doc2vec, Glove, LSI, LDA, CNN and other functions were mostly conducted with Python programming language and its libraries: gensim [3], nltk[4], sklearn[5], keras[6], libsvm[7]. R[8] and Weka [9] platforms were also used for cross validation, data preparation and some other analyses.

[3] radimrehurek.com/gensim/
[4] nltk.org
[5] scikit-learn.org
[6] keras.io
[7] github.com/cjlin1/libsvm
[8] r-project.org
[9] www.cs.waikato.ac.nz/ml/weka/

## 5. Results and Discussion

### 5.1. BoW

We report our experimental studies as shown in the four corresponding tables. The algorithms performance in the tables are in rounded F1-measure for the sake of simplicity. F1 measurement is the harmonic formula using recall and precision scores where precision is the ratio of correctly relevant instances to the retrieved instances, while recall is the ratio of correctly relevant instances to the total amount of relevant instances in the system. Table 1 shows scores of all machine learning algorithms across $\chi^2$ and IG selectional criteria under BoW representation. Apparently Information Gain and $\chi^2$ feature selection criteria performs similarly. Among the machine learning algorithms, the most successful one is found multi-NB classifier. It achieved 90 % F1 scores at the feature selection size of 4K. The table indicates that multi-NB classifier clearly outperforms other machine learning algorithms such as LR, DT, SVM and K-nearest neighboring (KNN). Term weighting metrics such as term frequency (tf) or inverse document frequency (idf) are widely-used techniques in document-term matrix in order to improve the performance in IR related problem, [21]. However, we do not observe any significant difference between weighting formula.

**Table 1.** BoW approach (F1 scores)

| Ft-Sel | NB | log Reg | SVM | KNN | DT |
|---|---|---|---|---|---|
| IG | **90** | 83 | 82 | 45 | 72 |
| $\chi^2$ | **90** | 81 | 82 | 44 | 71 |

### 5.2. Topic modeling

Table 2 indicates the performances of topic modeling paradigm. The dimension is set to 200 and 400 for both LSI and LDA. This table clearly suggests that LDA topic modeling showed poor performance. The dimension size does not improve its performance as well. LSI modeling highly outperformed LDA performance both in the size of 200 and 400. On the other hand we observed that logistic regression can show better performance as the dimension size increases for LSI. SVM showed better performance at even the size of 200. Other algorithms mostly underestimated the data.

**Table 2.** Topic Modeling ( F1 scores)

| Topic Model | Log. Reg. | SVM | KNN | DT |
|---|---|---|---|---|
| LSI + 200 Dim | 80 | 83 | 68 | 73 |
| LSI + 400 Dim | 84 | 84 | 58 | 51 |
| LDA + 200 Dim | 49 | 74 | 68 | 73 |
| LDA + 400 Dim | 49 | 69 | 65 | 69 |

### 5.3. Embeddings

Table 3 shows the performance of embedding models. There are five different document embedding models to be compared as shown in the table. Three models belongs to doc2vec model. The table indicates that document embeddings clearly outperformed the word embedding averaging approaches both in word2vec and glove models. PV-DM shows better performance and outperforms PV-BoW and other word averaging approaches. On the other hand, the concatenation of two training architectures does not contribute the performance. We do not observe any significant difference between PV-DM and concatenation (PV-DM+PV-DBoW). We conclude that the paragraph vector architecture has clearly showed better representation performance against word averaging. Among the classifiers, SVM showed slightly better performance among other ML algorithms. Logistic Regression is considered another successful algorithm.

**Table 3.** Embdeddings (F1 scores)

| Model | Log. Reg. | SVM | KNN | DT |
|---|---|---|---|---|
| Pv-DM + PV-BoW | 89 | 89 | 79 | 52 |
| PV-BoW | 87 | 88 | 79 | 50 |
| PV-DM | 89 | 88 | 77 | 53 |
| Word2vec Avg. | 81 | 84 | 72 | 52 |
| Glove Avg. | 81 | 86 | 73 | 53 |

### 5.4. The results of deep learning and final remarks

We tested fours deep learning algorithms by using keras python library: RNN, GRU, LSTM and CNN where LSTM and GRU are the special variants of RNN algorithm. Therefore the parameters of these three algorithms are almost same in the keras library. According to our experiment the most suitable dimensionality of the output space is 32, which is also called units. It means that there are 32 nodes in the hidden layer. Other parameters are experimentally selected depending on the model performance. The parameters are as follows: activation function is set to hyperbolic tangent, recurrent activation is set to hard sigmoid function, kernel initializer is set glorot uniform and recurrent initializer is set to orthogonal. CNN has a different architecture than the RNN and its variants. It has two layers: convolutional layer and pooling layer. For the convolutional layer we selected Conv1D where this layer is especially useful for the textual sequence where the dimensionality of the output space is set to 32 and kernel initializer is set to glorot uniform. For the pooling layer we selected the Max Pooling by using MaxPooling1D method in keras that is suitable for the textual data. Finally, Table 4 represents the performances of these deep Learning methods. It indicates that GRU highly outperformed other three methods. However, these deep learning models do not show better performance than the other approaches based on BoW and document embeddings.

Finally, when comparing the results of four different paradigms, we can observe that traditional BoW approach employing Multi-NB and IG Feature selection shares the same performance results with PV-DM of document embeddings. The differences in performance between these models are not statistically important in terms of t-test. Another interesting observation is that traditional

**Table 4.** Deep Learning (F1 scores)

| Model | F Score |
|---|---|
| CNN | 73 |
| LSTM | 75 |
| GRU | 81 |
| RNN | 75 |

BoW method clearly outperforms other NNLM based and embedding averaging methods. Therefore we conclude that although NNLM approaches has brought big advantages and movement to the field of NLP, traditional approaches such as bag-of-word representation with appropriate feature selection still have good capacity. [17] addressed that NNLM based paragraph model outperforms traditional models for the problem of sentiment analysis and the some IR related problems other than document classification. But for the Turkish text classification problem, paragraph vector and BoW approaches share the similar score.

In order to fairly compare the results with other studies, the comparison requires same configuration and same degree of difficulty. In Turkish, [4] achieved % 93 success rate for three classes genre detection. Each class consists of only 200 examples. [11] obtained at their best 95.8 % for six-class category detection where there is only 100 documents under each category. There exists another study whose configuration is roughly equal to that of our study where the number of classes is 6 and there exists 600 document for each category, [16]. They achieve % 90.1 and % 91.3 success rate with Random Forest and Zemberek Stemmer. The second model applied ARFS and obtained better results. When we use only 6 classes in our dataset, we got comparable results with them.

## 6. Conclusions

In this study, we applied different document representation approaches to Turkish document categorization. We categorized the recent studies under four different paradigms and applied them for document categorization in Turkish language. To give an equal comparison of these approaches, we prepared a benchmark dataset under seven document categories. The methods were tested within a supervised learning architecture, where popular machine learning techniques such as SVM or Naive Bayes were applied to the generated representations. We demonstrated that document embeddings models and traditional bag-of-words approaches achieved equally successful results. Although word embeddings, topic modeling, deep learning approaches have been successfully applied to word semantics, the document embeddings and traditional methods outperformed them for the document representation. On the other hand, interestingly, traditional BOW approaches still showed a comparable performance for the representation. Our architecture achieved successful results of 90 F1-score for the Turkish language.

## References

[1] B. Açıkalın and N. G. Bayazıt. 2016. The importance of preprocessing in Turkish Text classification. In *2016 24th Signal Processing and Communication Application Conference (SIU)*. 2053–2056. https://doi.org/10.1109/SIU.2016.7496174

[2] Burak Kerim Akkus and Ruket Çakıcı. 2013. Categorization of Turkish News Documents with Morphological Analysis. (2013).

[3] Mehmet Fatih Amasyalı, Sümeyra Balcı, Emrah Mete, and Esra Nur Varlı. 2012. A Comparison of Text Representation Methods for Turkish Text Classification. *EMO Scientific Journal* 2 (2012). arXiv:1309-5501

[4] M. Fatih Amasyalı and Banu Diri. 2006. Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. In *Natural Language Processing and Information Systems*, Christian Kop, Günther Fliedl, Heinrich C. Mayr, and Elisabeth Métais (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 221–226.

[5] Çağrı Toraman. 2011. *Text Categorization and Ensemble Pruning in Turkish News Portals*. Ph.D. Dissertation. Bilkent Uniiversity, Ankara.

[6] Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning Long-term Dependencies with Gradient Descent is Difficult. *Trans. Neur. Netw.* 5, 2 (March 1994), 157–166. https://doi.org/10.1109/72.279181

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937

[8] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR* abs/1406.1078 (2014). arXiv:1406.1078 http://arxiv.org/abs/1406.1078

[9] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41, 6 (1990), 391–407.

[10] M Fatih Amasyali, Aytunç Beken, and Yildiz Teknik Üniversitesi. 2018. Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sinifflandirmada Kullanilmasi Measurement of Turkish Word Semantic Similarity and Text Categorization Application. (03 2018).

[11] Aysun Güran, Selim Akyokus, Nilgün Güler Bayazıt, and M Zahid Gürbüz. 2009. (07 2009).

[12] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

[13] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. (2001).

[14] I.T. Jolliffe. 2002. *Principal Component Analysis*. Springer.

[15] Daniel Jurafsky and James H. Martin. 2016. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[16] Deniz Kılınç, Akın Özçift, Fatma Bozyigit, Pelin Yıldırım, Fatih Yücalar, and Emin Borandag. 2017. TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science* 43, 2 (2017), 174–185. https://doi.org/10.1177/0165551515620551 arXiv:https://doi.org/10.1177/0165551515620551

[17] Quoc Le and Tomas Mikolov. [n. d.]. Distributed Representations of Sentences and Documents. In *In NAACL HLT*. 2013.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (27 May 2015), 436–444. https://doi.org/10.1038/nature14539

[19] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. 2278–2324.

[20] David D. Lewis and Marc Ringuette. 1994. A Comparison of Two Learning Algorithms for Text Categorization. In *In Third Annual Symposium on Document Analysis and Information Retrieval*. 81–93.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

[24] G. Salton. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[25] Hinrich Schutze, David A. Hull, and Jan O. Pedersen. 1995. A comparison of classifiers and document representations for the routing problem. In *ANNUAL ACM CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL - ACM SIGIR*. ACM, 229–237.

[26] Jonathan M Cheek Stephen R Briggs. 2007. The role of factor analysis in the development and evaluation of personality scales. (2007).

[27] P. Tüfekci, E. Uzun, and B. Sevinç. 2012. Text classification of web based news articles by using Turkish grammatical features. In *2012 20th Signal Processing and Communications Applications Conference (SIU)*. 1–4. https://doi.org/10.1109/SIU.2012.6204565

[28] D. Torunoğlu, E. Çakirman, M. C. Ganiz, S. Akyokuş, and M. Z. Gürbüz. 2011. Analysis of preprocessing methods on classification of Turkish texts. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*. 112–117. https://doi.org/10.1109/INISTA.2011.5946084

[29] Filiz Türkoğlu, Banu Diri, and M. Fatih Amasyalı. 2007. Author Attribution of Turkish Texts by Feature Mining. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, De-Shuang Huang, Laurent Heutte, and Marco Loog (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1086–1093.

[30] Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information Processing and Management* 50, 1 (2014), 104 – 112. https://doi.org/10.1016/j.ipm.2013.08.006

[31] Savaş Yildirim. 2014. A Knowledge-Poor Approach to Turkish Text Categorization. In *Computational Linguistics and Intelligent Text Processing*, Alexander Gelbukh (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 428–440.

[32] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative Study of CNN and RNN for Natural Language Processing. *CoRR* abs/1702.01923 (2017). arXiv:1702.01923 http://arxiv.org/abs/1702.01923