


ITC 3/51 Information Technology and Control Vol. 51 / No. 3 / 2022 pp. 429-445 DOI 10.5755/j01.itc.51.3.29907	A Comprehensive Study of Learning Approaches for Author Gender Identification	
	Received 2021/09/29	Accepted after revision 2022/01/12
	 <a href="http://dx.doi.org/10.5755/j01.itc.51.3.29907">http://dx.doi.org/10.5755/j01.itc.51.3.29907</a>	

**HOW TO CITE:** Dalyan, T., Ayril, H., Özdemir, Ö. (2022). A Comprehensive Study of Learning Approaches for Author Gender Identification. *Information Technology and Control*, 51(3), 429-445. <http://dx.doi.org/10.5755/j01.itc.51.3.29907>

# A Comprehensive Study of Learning Approaches for Author Gender Identification

Tuğba Dalyan, Hakan Ayril, Özgür Özdemir

Department of Computer Engineering; Istanbul Bilgi University; Istanbul, Turkey

Corresponding author: [ozgur.ozdemir@bilgi.edu.tr](mailto:ozgur.ozdemir@bilgi.edu.tr)

In recent years, author gender identification is an important yet challenging task in the fields of information retrieval and computational linguistics. In this paper, different learning approaches are presented to address the problem of author gender identification for Turkish articles. First, several classification algorithms are applied to the list of representations based on different paradigms: fixed-length vector representations such as Stylometric Features (SF), Bag-of-Words (BoW) and distributed word/document embeddings such as Word2vec, fastText and Doc2vec. Secondly, deep learning architectures, Convolution Neural Network (CNN), Recurrent Neural Network (RNN), special kinds of RNN such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), C-RNN, Bidirectional LSTM (bi-LSTM), Bidirectional GRU (bi-GRU), Hierarchical Attention Networks and Multi-head Attention (MHA) are designated and their comparable performances are evaluated. We conducted a variety of experiments and achieved outstanding empirical results. To conclude, ML algorithms with BoW have promising results. fast-Text is also probably suitable between embedding models. This comprehensive study contributes to literature utilizing different learning approaches based on several ways of representations. It is also first attempt to identify author gender applying SF on Turkish language.

**KEYWORDS:** Author gender identification, Stylometric features, Deep learning, Embeddings.

---

## 1. Introduction

In recent years, authorship analysis has attracted considerable attention and a number of techniques are formulated to address this fundamental challenge. Authorship profiling that is one of the core tasks in authorship analysis, is related to determining authors' personality type, age and gender. The author gender identification has been seen as a subproblem of the authorship profiling and aims to assign documents to one of the author genders. Advances in author gender identification have raised interest in various fields including forensics, security, e-mail forgery, on-line communities, security, trading and marketing, etc. but it is also applicable to academic fields such as information retrieval and computational linguistics. (e.g. PAN is a series of scientific events and shared tasks such as authorship attribution, obfuscation evaluation, authorship verification, gender prediction, etc.)

Similar to text classification problem, the important point is feature extraction that resolves the issue of representing a document as a feature vector in author gender identification problem. A simple and traditional technique is Bag-of-Words (BoW) where each feature corresponds to a word or token based on a metric such as word frequency. Another widely used feature representation relies on the writing style of the author, which can be characterized through stylometric features (SF). The approach is based on the assumption that each author has a characteristic and unique stylistic tendency. These author-related features are generally categorized into five groups: character-based, syntactic, word-based, structure-based and function words based. These automatically extracted features are composed as a fixed length sequence of vectors and fed to machine learning algorithms in order to determine the author gender.

Traditionally, although the fixed-length vector representations are used as the state-of-the-art in various NLP applications, they have some drawbacks such as high dimensionality, sparsity, etc. Therefore, considerable efforts have been devoted to continuous space model (distributed word embedding), which involves distributed feature learning over sequences of words/tokens and it has effectively dominated several NLP tasks. Many training approaches have been proposed and the pre-trained word embeddings have been

found to be good at extracting semantic and syntactic regularities [10, 19, 58, 41]. Word2vec [41, 40] is one of the efficient models for learning word embeddings through CBoW and SG architectures using neural networks. FastText [12, 30] is a simple and efficient model that allows users to learn text representations as embeddings. Doc2vec [36] applies unsupervised learning to infer continuous representations for larger blocks of text. Moreover, these representations are fed into Deep Neural Network (DNN) architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), which became popular and have revolutionized the way to address various NLP tasks [68].

In this study, different learning approaches are proposed to address author gender identification for Turkish articles. First, eleven different classifiers are employed over fixed-length features that are obtained by SF and BoW, and distributed embeddings, specifically Word2vec, fastText and Doc2vec. Then, CNN, RNN and special types of RNN such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), C-RNN, Bi-directional LSTM (bi-LSTM), Bi-directional GRU (bi-GRU), Hierarchical Attention Networks and Multi-head Attention (MHA) are trained, and their performances are evaluated. The empirical results show that traditional methods outperform state-of-the-art of deep learning methods for author gender identification problem, and fastText has the best performance among embedding models. This study is considered to be the first comprehensive study employing stylometric features to address the author gender identification problem for the Turkish Language. Moreover, the empirical results are compared with the results achieved by different embeddings and DNN architectures.

---

## 2. Related Work

The author gender identification is one of the well-studied tasks in NLP domain. Most of these studies utilize features such as words, n-grams, Part of Speech (POS) tags, etc. In [50], Schler et al. consider the impact of using style-related (POS tags, function words and blog-specific features) and content-related

features, on age and gender information, for blogging based data. In [43], a supervised approach based on POS sequence patterns is proposed for gender classification of blog authors; they also present a feature selection method based on ensemble feature selection. There are some other studies which present gender classification of blog entries using different classes of features such as SF, gender preferential features, word classes, etc. [44, 64, 8]. In [33], Koppel et al. use a combination of function words and POS to infer the author gender of British National Corpus (BNC) with 80% accuracy.

In [16], Cheng et al. present a model to identify the e-mail authors' gender, they carry out training of a Support Vector Machine (SVM) and a Decision Tree (DT) using 545 features based on character, word, syntax, structure and function words. They also introduce psycholinguistic and gender-linked features along with SF. They indicate that the SVM method outperforms the DT method, and function-word-based, and word-based features are crucial for gender identification. In [15], Cheng et al. address author gender identification for short length, content-free and multi-genre text. Three classifiers (SVM, Bayesian Logistic Regression, AdaBoost) were designed and SVM outperformed the others with 76.75% and 82.23% accuracy on two different datasets (Reuters and Enron Corpus).

Burger et al. [14] proposes a study on identifying gender of Twitter entries using a number of text-based features and several different classifiers, including Balanced Winnow (BW), NB and SVM. Similarly, Deitrick et al. [20] employ a Modified Balanced Winnow (MBW) classifier on the author gender identification task by using the Enron email dataset. Moreover, they introduce a set of SF and compare them with word-count features. The empirical studies resulted that word-count features outperform the introduced set of SF. In this study [20], the classifiers showed sensitivity to parameters, thereby proving the requirement of optimal parameter tuning on SF. In a subsequent study, Deitrick et al. [21] showed that exploiting feature selection methods improves the results substantially on n-gram features. Several studies [5, 6, 37, 39, 42, 47, 59, 61] also presented models for author gender detection of writers of the social media in different languages.

Alsmearat et al. [2] utilize conventional BoW features by applying feature selection and reduction methods

on the author gender identification task for the Arabic language. In experimental results, the Stochastic Gradient Descent (SGD) showed superiority among several machine learning algorithms by achieving 94.1% accuracy on the dataset collected from Arabic news articles. Subsequently, Alsmearat et al. [3] examine the emotions on author gender identification, however, the experiments did not show any significant effect of emotions on this task. Later, Alsmearat et al. [3] extended their SF set to 363 features and surveyed the proposed set compared to BoW features. The empirical studies showed that having a larger set of SF outperformed BoW representations in the Arabic language. Besides, they reported that using dense stylistic features is more efficient against BoW due to the feature dimension.

In recent studies, the dense and low-dimensional real-valued vectors have been found to be effective for many NLP tasks due to the problems of traditional methods such as high dimensionality, sparsity, etc. In [9], Bayot et al. use the averages of word embeddings as features, specifically Word2vec and SVM are trained to address author gender and age classification problem. Using the PAN 2016 dataset, they achieve 44.8% and 68.2% accuracy for age and gender classification in English, respectively. In [38], Markov et al. use Doc2Vec to train a Logistic Regression (LR) classifier on the PAN author profiling 2014-2016 corpora. One task of PAN 2018 is to address gender identification from texts but also from images for languages Arabic, English, and Spanish.

In studies [48, 49], two different approaches based on ML and CNN are proposed for automatic text classification problem for the author gender in the Russian-language. CNN obtained an accuracy of 86%. In [32], Kodiyan et al. propose a bidirectional RNN architecture implemented with an attention-based GRU; and they obtain an accuracy of 75.31% in gender classification. Also, in [13] an LSTM architecture is employed to address the problem of gender identification.

In Turkish, the study conducted by Amasyalı et al. [7] utilized n-gram representations on the author gender identification task. Moreover, they used Correlation-based Feature Selection (CFS) to select subsets of the features. In the experiments, SVM showed the best results by achieving 96.3% accuracy on the dataset collected from Turkish newspaper articles. Talebi et al. [55] use NB, SVM and KNN as classifier

on Facebook comments to identify gender, age and education level. They show that NB classifier gave an accuracy of 90.85% for gender, 89.67% for age and 86.15% for education level. Yıldız [68] compared the BoW representations with low-dimensional real-valued vectorization approaches on the author gender identification task in Turkish news articles. The results showed that using BoW features selected by the chi-squared statistics outperformed Word2Vec, Doc2Vec and GloVe embeddings by achieving 91% F1-score.

## 3. Methodology

### 3.1. Dataset

In this study, we extended the dataset used in [68]. The dataset contains news articles written in the Turkish language and has an even distribution in terms of authors' gender. The further details are given in Table 1.

**Table 1**

Summary of the dataset

Features	Male	Female
# of authors	145	145
# of articles	10864	10864
avg # of sentences / article	34.74	43.00
avg # of words / article	575.91	624.66
avg # of characters / article	4475.01	4837.28
avg # of characters / sentences	128.78	112.47
avg # of characters / word	7.77	7.74
avg # of words / sentences	16.57	14.52

Several preprocessing operations are conducted in order to extract stylometric features by using Natural Language Toolkit (NLTK)<sup>1</sup>. We applied lem-

matization and POS Tagging in order to extract the word-based features and features based on function words in stylometric analysis. The number of nouns, adjective, verb, etc. in text may be important features to determine the author gender. All these POS tags are used as function words that are listed in Table 2. Furthermore, the lemma of words is used in formulas to obtain the value of word-based features. To see the importance of these features, all texts are lemmatized by means of a morphological parser. For each given word, the process of classifying words into their POS and lemmatization is done as in the following example. We utilized a morphological parser which is based on a two-level morphology with an accuracy of 98% and an averaged perceptron-based morphological disambiguator for Turkish [46]. English form of experiments is conducted with NLTK.

**Turkish:** O ve annesi dün gece güldüler.

**English:** She and her mum laughed in last night.

**POS-Turkish:** o\*det ve\*conj annesi\*noun dün\*noun gece\*noun güldüler\*verb . \*punc

**POS-English:** [(‘She’, ‘PRP’), (‘and’, ‘CC’), (‘her’, ‘PRP’), (‘mum’, ‘NN’), (‘laughed’, ‘VBN’), (‘in’, ‘IN’), (‘last’, ‘JJ’), (‘night’, ‘NN’), (‘.’, ‘.’)]

**Turkish:** o anne ve dün gece gül

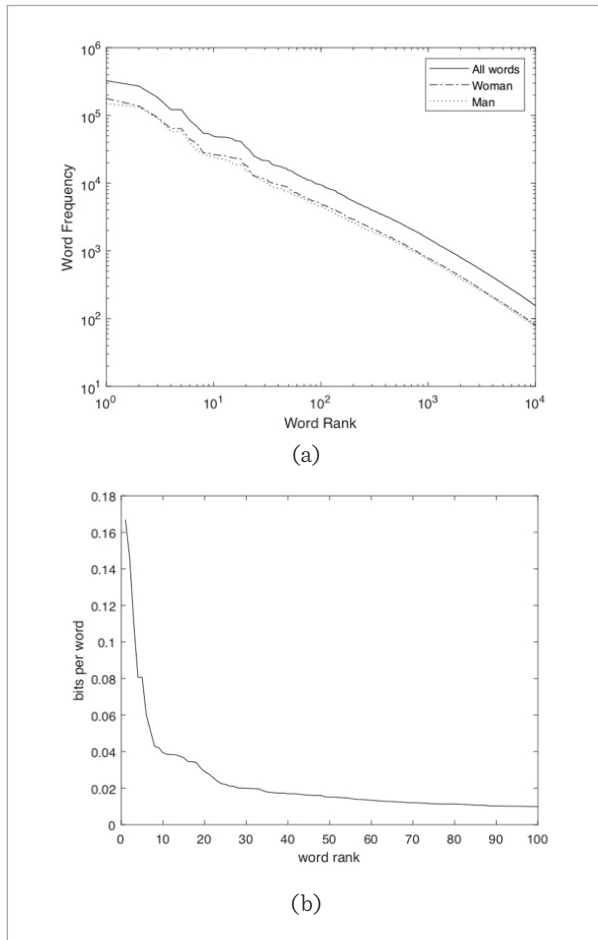
**English:** she and her mum laugh last night

The rank-frequency distribution of the words on the dataset is almost linear at the log-log scale (Figure 1a), conforming to the Zipf’s law as expected from a natural language; the same is also valid for both the male and female author subsets. Before stemming/lemmatization and stop word removal, but after the punctuation removal and conversion to lower case, the average information content of each word on the dataset is computed to be 11.2633 bits/word for the whole corpus. Figure 1b shows the contribution of the most frequent 100 words to the average entropy per word, the distribution computed for the male and female authored data subsets have values very close to the average, in the order of  $\pm 0.01$  bits per word both for the average and for the individual contributions of the most frequent words; this is another indicator of homogeneity and class balance of the dataset, not just at the document level but also at the word level.

<sup>1</sup> [www.nltk.org](http://www.nltk.org)

**Figure 1**

Word distributions and information entropy: (a) Word rank vs frequency for dataset, (b) Contribution of most frequent words to average information entropy



## 3.2. Experiment Setup

### 3.2.1. Stylometric Features (SF)

The Stylometric Features (SF) extract features based on writing styles of authors of different genders. We captured a large number of SF and classified them into five sets: (1) character-based; (2) word-based; (3) syntactic; (4) function words; and (5) structure-based. Table 2 lists all the features with descriptions.

Character-based features are one of lexical features [1]. They represent the frequency of individual characters. In this study, we employed 37 features such

as number of characters, letters, special characters, tab space, etc. Syntactic features concern with writing formation patterns such as regular punctuations (comma, colon, period, etc.) and irregular ones (ellipsis, exclamation marks, etc.) which are particular to author. We introduce 11 syntactic features as listed in Table 2. Another group of features is structure-based features, which are another strong authorial evidence of writing style. The features such as number of lines, sentences, paragraphs, etc. are also considered and 12 features of this type are utilized as structure-based features. Word-based features represent features related to number of words, average length per word, etc. 13 word-based features are extracted. Besides that, we exploit some vocabulary richness measures such as: Yule, Simpson, Sichel, Honore, Entropy, Hapax dislegomena and Hapax legomena as features. The formulas are given in Table 3 where  $V$  is number of different words,  $V_i$  is number of different words that occur  $i$  times, and  $N$  is total number of words. Function words are grammatical words that are related to the use of grammatical relations with other words. We captured 14 grammatical relation-based features.

Some of these features are transferred from previous studies [15, 3]. In addition, we categorized 30 topics such as sport, politics, botanic, etc. and created a vocabulary for each category. For example, we listed the words such as volleyball, training, trainer for sport category. For each document, we calculate how many terms are included in such categories. We use the count of these function words as features. Besides that, some features are particular to Turkish language such as diacritics. Although these characters are not seen in text frequently, it may show the characteristic tendency of gender. Emotion expressing words or symbols (hi, emojis, :, |, etc.) allow to describe situations, feelings, objects in communication via Twitter, Instagram, etc. So, we exploit such emotion words and symbols as features. While the experiments in [4] showed that there is no significant evidence for accuracy improvement due to usage of emotion/sentiment features, studies [16, 15] propose that the tendency of female authors' writings are more emotional than male authors. They introduce 9 gender-linked features such as lovely, quite, really, etc. In [3], Alsmearat use the number of apologetic words and feminine words to indicate that the female authors might use

**Table 2**

The proposed feature set and descriptions

Feature	Feature Description	Feature	Feature Description
<b>Char.</b>		<b>Word</b>	
Ft1	Tot. num. of characters (C)	Ft61	Tot. num. of words (W)
Ft2	Tot. num. of letters(a-z)/C	Ft62	Tot. num. of repeated word/W
Ft3	Tot. num. of lower characters/C	Ft63	Tot. num. of short words (1-3 characters)/W
Ft4	Tot. num. of upper characters/C	Ft64	Tot. num. of words longer than 6 characters/W
Ft5	Tot. num. of digits/C	Ft65	Avg. length per word (in characters)
Ft6	Tot. num. of white-space characters/C	Ft66	Vocabulary richness (total different words/W)
Ft7	Tot. num. of tab space characters/C	Ft67	Hapax legomena/W
Ft8	Tot. num. of special characters/C	Ft68	Hapax dislegomena/W
Ft9	Tot. num. of emoji/C	Ft69	Yule's K measure
Ft10	Tot. num. of positive emojis/C	Ft70	Sichel's S measure
Ft11	Tot. num. of neutral emojis/C	Ft71	Honore's R measure
Ft12	Tot. num. of negative emojis/C	Ft72	Simpson's D measure
Ft13	Tot. num. of diacritics/C	Ft73	Entropy measure
Ft14-37	Num. of special characters (% , & , etc.)/C		
<b>Syntactic</b>		<b>Funct. words</b>	
Ft38	Tot. num. of single quotes (')/C	Ft74	Tot. num. of pro-sentence words/W
Ft39	Tot. num. of commas (,)/C	Ft75	Tot. num. of nouns/W
Ft40	Tot. num. of periods (.) /C	Ft76	Tot. num. of adjectives/W
Ft41	Tot. num. of colons (:)/C	Ft77	Tot. num. of adverbs/W
Ft42	Tot. num. of semi-colons (;)/C	Ft78	Tot. num. of conjunctions/W
Ft43	Tot. num. of question marks (?)/C	Ft79	Tot. num. of determiners/W
Ft44	Tot. num. of exclamation marks (!)/C	Ft80	Tot. num. of duplications/W
Ft45	Tot. num. of multiple question marks (???) /C	Ft81	Tot. num. of interjections/W
Ft46	Tot. num. of multiple exclamation marks (!!!) /C	Ft82	Tot. num. of questions/W
Ft47	Tot. num. of ellipsis (...) /C	Ft83	Tot. num. of verbs/W
Ft48	Tot. num. of double quotes (" )/C	Ft84	Tot. num. of prepositions/W
		Ft85	Tot. num. of numbers/W
		Ft86	Tot. num. of pronouns/W
		Ft87	Tot. num. of punctuations/W
		Ft88-117	Tot. num. of categories/W (30 features)
<b>Structure</b>			
Ft49	Tot. num. of lines		
Ft50	Tot. num. of sentences (S)		
Ft51	Tot. num. of paragraphs		
Ft52	Num. of sentences beginning with upper case/S		
Ft53	Num. of sentences beginning with lower case/S		
Ft54	Avg. num. of words per paragraph		
Ft55	Avg. num. of characters per paragraph		
Ft56	Avg. num. of sentences per paragraph		
Ft57	Avg. num. of words per sentence		
Ft58	Tot. num. of blank lines/Tot. num. of lines		
Ft59	Avg. length of non-blank line		
Ft60	Absence/present of greeting words		

**Table 3**

Vocabulary richness measures

Measures	Formulas
Yule's K measure	$K = 10^4 \left( -\frac{1}{N} + \sum_{i=1}^V V_i \left( \frac{i}{N} \right)^2 \right)$
Simpsons's D measure	$D = \sum_i^V V_i \frac{i-1}{N(N-1)}$
Sichel's S measure	$S = \frac{\text{countofHapaxLegomena}}{V}$
Honore's R measure	$R = \frac{100 \log_{10} N}{1 - \frac{\text{countofHapaxLegomena}}{V}}$
Entropy measure	$E = \sum_i^V V_i \left( -\log_{10} \frac{i}{N} \right) \frac{i}{N}$
Hapax Dislegomena	Words that occur only twice
Hapax Legomena	Words that occur only once

more apologetic and feminine words than male authors. We did not cover such features that are based on gender specific words. We also exploit pro-sentence words and greeting words as features. All these additional features that are based on mostly Turkish language are given in Table 4. We propose 117 SF in 5 categories to build the feature space. The experiments with classifiers and feature selection process have been performed using Python scikit-learn<sup>2</sup> and Tensorflow Keras<sup>3</sup> libraries.

**Table 4**

Features and some of examples

Features	Examples
Diacritics	â, ê, î, ô, û, Â, Ê, Î, Ô, Û
Positive emojis	:), :D
Neural emojis	: , :0
Negative emojis	:(, <
Pro-sentences	yes, no, ok
Greetings	thanks, hi
Categories	sports, politics, health, technology, botanic

<sup>2</sup> <http://scikit-learn.org/stable/>

<sup>3</sup> [https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras)

### 3.2.2. Bag-of-Words Features (BoW)

Traditionally, the Bag of Words (BoW) or Bag of n-grams is used as the state-of-the-art feature vector representation in many NLP tasks. Each document is represented as an unordered set of features that correspond to the terms in a vocabulary, which could be words, token/character n-grams, for a document collection. Each term in a feature vector is represented by a numeric value, the value can be a count or the value is calculated in different measures such as tfidf. In this study, we computed weighted form of BoW with bi-grams using tf-idf (BoW-tfidf).

We limited the size of vocabulary to 15K words for the performance of the n-fold (n=10) cross validation. One of the drawbacks of BoW model is the high dimensionality. The most common way to address this issue is to remove auxiliary terms such as stop words or to determine a frequency threshold to reduce the size of vocabulary. In our experiments, we chose the latter and removed the words occurs less than 5 times from our vocabulary.

Furthermore, several dimension reduction methods are applied to reduce the size of feature space in addition to frequency thresholding, such as Principal Component Analysis (PCA), Non-negative matrix factorization (NMF), Random Projection (RP), t-Distributed Stochastic Neighbor Embedding (tSNE), and auto-encoder using Python scikit-learn and Tensorflow Keras libraries. We eliminated Linear Discriminant Analysis (LDA) for our BoW experiments, because both LDA and tf-idf techniques are developed to represent inter and intra class information. Although some studies [72] combine these two techniques with using Word2vec representations, we didn't implement such architecture for simplicity of our experiments and comparisons.

### 3.2.3. Embeddings

The fixed-length vector representations for documents have some drawbacks such as high dimensionality, sparsity, loss of positional information and lack of semantic encapsulation [34]. So, distributed vector representation has recently become very popular to overcome the weakness of traditional feature representations. In this study, Word2vec and fastText are utilized to capture feature vectors, with window size set to 10. We run the experiments with using CBoW with Negative Sampling (NS).

In addition to word embeddings, document embeddings such as Doc2vec, also named paragraph vectors,

can represent collections of many words such as sentences, paragraphs, and documents. A neural network architecture is trained to produce 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 added to head position of each context window. The architecture predicts the last word using all vectors of the words in the context plus the paragraph vector. In this study, we run the experiments using Paragraph-Vector Distributed Memory (PV-DM). The experiments are conducted using Python, scikit-learn, and Gensim<sup>4</sup>, NLTK libraries and fastText<sup>5</sup>.

### 3.2.4. Deep Neural Networks (DNN) Architectures

Neural networks with the pre-trained word vectors play an important role in many NLP tasks [19, 68, 31, 54, 53, 62]. In this study, we compared the conventional machine learning algorithms with deep learning approaches. We employed four different deep learning structures with different variants namely, CNN, RNN, C-RNN, Hierarchical Attention Networks [65], and Multi-head Attention (MHA) [60].

In CNN architecture, let  $x_i$  be the  $k$ -dimensional word embedding vector corresponding to the  $i$ -th word in the sentence. A convolution operation with a filter  $w$  is applied to sentence vector  $x$ . A feature  $c$  is produced as following operation

$$c_i = f(w \odot x_{i:i+h-1} + b), \quad (1)$$

where  $f$ ,  $\odot$ ,  $b$ ,  $h$  is ReLu activation function, convolution operation, bias term and window size respectively. Feature map at  $j$ -th window is formed as

$$c_j = [c_j, c_{(j+1)}, c_{(j+2)}, \dots, c_{(n-h+1+j)}], \quad (2)$$

where  $n$  is the sentence length. In order to extract the significant features on feature map  $c$ , each  $j$ -th window response is down-sampled by a max-pooling operation [19]. Then, the maximum feature map  $c'$  is passed to a fully connected dense network in order to estimate the output.

<sup>4</sup> <https://radimrehurek.com/gensim/>

<sup>5</sup> <https://github.com/facebookresearch/fastText>

RNNs are widely used deep learning approaches for NLP problems, due to their architecture which is designed to tackle sequence modelling and temporal dependencies [26]. In this architecture, the network aims to map a given input sequence  $x = (x_1, \dots, x_t)$  to a hidden vector sequence  $h = (h_1, \dots, h_t)$ .

The output vector sequence  $y = (y_1, \dots, y_t)$  is then calculated according to Equations (3) and (4).

$$h_t = f(x_t W_{xh} + h_{(t-1)} W_{hh} + b_h) \quad (3)$$

$$y_t = g(h_t W_{hy} + b_y), \quad (4)$$

where  $x_t$  is the input at time step  $t$ ;  $h_t$  is the hidden vector at time  $t$ ;  $W$  is weight matrix that models  $W_{xh}$  as input-to-hidden,  $W_{hh}$  as hidden-to-hidden (recurrent) and  $W_{hy}$  as hidden-to-output connections;  $b_h$  represents bias term for the hidden layer and by is the bias term for output layer.  $f(\cdot)$  and  $g(\cdot)$  are nonlinear activation functions such as sigmoid or tanh.

However, simple RNNs face difficulties to capture long-term dependencies because of vanishing or exploding gradient [11, 28] problem. One of the solutions to deal with the vanishing and exploding gradient problem is using gating mechanism to control the information flow from previous hidden layers [29]. The popular recurrent gating units, namely LSTM and GRU, proved themselves on various NLP tasks with robust results [69].

Long-short Term Memory (LSTM) [29] is one of the variants of RNN architecture that aims to control the existing memory and omitting the unrelated information and memory cells to store the information across time [25]. The architecture of LSTM use memory cells which have linear dependence on its current activity ( $c_t$ ) and its past activity ( $c_{(t-1)}$ ) [66]. The information flow between the past and the current activities is modulated by using a forget gate. The stored information in memory cell is calculated as follows

$$c_t = f_t * c_{(t-1)} + i_t * g_t \quad (5)$$

$$h_t = o_t * \tanh(c_t), \quad (6)$$

where it is input,  $f_t$  is forget,  $o_t$  is output gates and  $g_t$  is the vector of memory cell updates,  $c_t$  is memory cell at time  $t$ ,  $h_t$  is the hidden state vector at time  $t$ , and  $*$  is the element-wise multiplication [52, 24, 57].

Gated Recurrent Unit (GRU) is another variant of RNNs, proposed to simplify the architecture of LSTM



by reducing the number of gates [17]. Instead of triple gate  $(i_t, f_t, o_t)$  structure of LSTM, GRU uses two gates, i.e. reset gate  $r_t$  and update gate  $z_t$ . The hidden state vector  $h_t$  is then computed by

$$h'_t = \tanh(W[r_t * h_{(t-1)}, x_t]) \quad (7)$$

$$h_t = (1 - z_t) * h_{(t-1)} + z_t * h'_t. \quad (8)$$

In this study, we employed the given three RNN variants in forms of unidirectional and bidirectional [51] structures. Apart from the above-mentioned unidirectional structures, bidirectional RNN proposes to feed sequences on both forward and backward direction of given time series. The output  $y_t$  is computed by

$$y_t = g(h_t^{\leftrightarrow} W_{hy}^{\leftrightarrow} + b_y^{\leftrightarrow}), \quad (9)$$

where  $h_t^{\leftrightarrow}$  denotes that the calculated hidden state vector  $h_t = [h_t^{\leftarrow}; h_t^{\rightarrow}]$  with respect to forward and backward direction of sequence;  $W_{hy}^{\leftrightarrow}$  and  $b_y^{\leftrightarrow}$  follows same notion.

C-RNN, originally proposed as C-LSTM [71], combines CNN and RNN architectures in order to learn both word representations and sequences information. Some studies [35, 48, 49] also employ the combinations CNN+LSTM or Recurrent Convolutional Neural Networks to address the limitation of the RNN and CNN models for text classification. We also used such structures to see whether it contributes the accuracy.

In C-RNN architecture, word embedding vector is used for computing the feature  $c$ , as in Equation (1), then feature map  $c' = \max\{c_j\}$  is formed as in Equation (2). Instead of feeding maximum feature map vector to fully connected dense network directly, feature maps pass through an RNN architecture. We experimented with Basic-RNN, LSTM and GRU for RNN part of C-RNN and concluded that the most successful model is LSTM based on experimental results for our task. For sake of correctness of the terminology, we will call our LSTM based C-RNN implementation as C-LSTM for the rest of the paper.

Hierarchical Attention Network is proposed to learn word and sentence level representations of documents by using two distinct attention layers [65]. The architecture consists of two RNN architectures connected in a sequential manner. The first component of the hierarchical network is word-level RNN structure similar to the above-mentioned RNNs. However, the

output is passed through another attention layer before being fed to a fully connected dense network. The word attention vectors  $s_w$  are computed by

$$u_t = \tanh(W_w h_t + b_w) \quad (10)$$

$$\alpha_t = \text{softmax}(u_t^T u_w) \quad (11)$$

$$s_w = \sum \alpha_t h_t, \quad (12)$$

where  $W_w$  and  $b_w$  are weights and bias term of word representations; and  $h_t$  is the hidden state at position  $t$ . Then, word attention vector  $s_w$  is fed to fully connected dense network. The latter component of the hierarchical network, i.e. sentence-level RNN structure, is fed by word attention vectors map  $[s_w^1, s_w^2, \dots, s_w^L]$  in order to compute sentence attention vectors  $s_s$  as in Equation (12). The sentence attention vectors  $s_s$  are then passed through the fully connected dense network which learns to classify the document based on sentence level attention output. Although original paper proposes the use of GRU, we also experimented with LSTM architecture at both word and sentence level RNN structure.

Another attention mechanism, namely Multi-head Attention (MHA), proposed by Vaswani et al. [60] achieved good results in various problems in NLP domain [27, 22]. MHA comprises of multiple attention heads that contain query (Q), key (K) and value (V) vectors to map the given query to an attention vector that is obtained by

$$\text{attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (13)$$

where  $d_k$  is the dimension of the keys. Following this, the attention heads are concatenated to project the feature vector by

$$MHA(Q, K, V) = \text{concat}(H_1, \dots, H_n) W^o, \quad (14)$$

where  $H_i$  refers to attention output of  $i$ -th head (Equation 10), and  $W^o$  represents the weight vector for attention heads. The rest of the architecture is similar to the models mentioned above. That is, the feature vector calculated by Equation (11) is passed to fully connected dense network.

We ran several independent experiments to determine the hyperparameters of each model. Consequently, we set the reduced dimension to 32 and 256

for respectively SF and embedding experiments, in the experiments involving feature extraction. Similarly, we trained the auto-encoders having 3-hidden layers that input and output layers have respectively the same unit size and the middle layer has the halves. In ML experiments,  $k$  was set to 5 for the kNN model; radial basis kernel was used for the SVM model; the kNN classifier that  $k$  is set 5 was employed for Bagging. In our DNN experiments, we chose 64 as the number dimensions for word embedding. In CNN experiments, the architecture proposed by Zhang et al. [70] is employed. That is, the input embedding is passed to CNN layer to map 3 different sized regions. The resultant feature vectors of different regions are combined and forwarded to a fully connected dense network. For RNN experiments, 3-stacked RNN architecture with decreasing unit size from 128 to 32 is utilized. We set attention vector dimension to be 128 for hierarchical attention network. In MHA experiments, 3-stacked network that each stack owns 4-headed MHA is employed. For fully connected dense network, 2 dense layers with respectively 128 and 64 neurons are used followed by 1 softmax layer. For reproducibility of our experiments, we share our codes<sup>6</sup>.

**Table 5**

The results in F1-score based on Stylometric Features.

Note. When tf-idf vectors are used, Nearest Centroid is also known as Rocchio classifier and AE refers for auto-encoder

Classifier	SF-all	PCA	LDA	NMF	RP	t-SNE	AE
k-NN	0.67	0.67	0.67	0.58	0.66	0.58	0.52
NB	0.62	0.61	0.72	0.59	0.47	0.50	0.63
SVM	0.59	0.59	0.72	0.44	0.58	0.60	0.42
DT	0.70	0.66	0.62	0.57	0.61	0.56	0.51
RF	<b>0.80</b>	0.77	0.62	0.62	0.68	0.58	0.51
SGD	0.74	0.52	0.72	0.14	0.51	0.56	0.51
LR	0.66	0.58	0.72	0.29	0.57	0.58	0.52
Boosting	0.51	0.73	0.73	0.60	0.64	0.55	0.52
Bagging	0.37	0.66	0.67	0.57	0.66	0.51	0.45
NC*	0.58	0.51	0.72	0.11	0.48	0.55	0.52
MLP	0.44	0.47	0.73	0.46	0.48	0.57	0.49

<sup>6</sup> [https://github.com/ozgurozdemir/author\\_gender\\_identification](https://github.com/ozgurozdemir/author_gender_identification)

**Table 6**

The results in F1-score based on BoW-tfidf

Classifier	BoW-tfidf only	PCA	NMF	RP	t-SNE	AE
k-NN	<b>0.87</b>	0.87	0.77	0.59	0.77	0.53
NB	0.77	0.73	0.75	0.60	0.58	0.61
SVM	0.75	0.73	0.67	0.61	0.76	0.68
DT	0.67	0.71	0.71	0.53	0.73	0.52
RF	0.81	0.84	0.80	0.57	0.76	0.53
SGD	0.85	0.78	0.66	0.59	0.73	0.61
LR	0.85	0.77	0.72	0.60	0.77	0.53
Boosting	0.75	0.79	0.75	0.59	0.57	0.63
Bagging	<b>0.87</b>	0.86	0.77	0.59	0.52	0.66
NC	0.73	0.71	0.71	0.60	0.57	0.63
MLP	0.55	0.54	0.51	0.51	0.73	0.49

## 4. Experiment Result

Several experiments are conducted to analyze the performance of classifiers across different feature vectors, with and without dimension reduction techniques. In the experiments, we used 10-fold cross-validation by reserving 20% of the dataset to test for each iteration. We utilized a list of classifiers: k-NN (k-Nearest Neighbours), Naive Bayes (NB), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), Stochastic Gradient Descent (SGD), Logistic Regression (LR), Boosting, Bagging, Nearest Centroid (NC), Multi-layer Perceptron (MLP). We measured the performances of these models in terms of accuracy, F1, recall, and precision. Since the dataset used in the experiments are evenly distributed, we reported only the F1-score for the brevity of the comparison. While BoW and embedding features capture a universal representation of the text, SF preserves the author's identity and provides a denser representation. For each news article, we extracted 117 SF to prepare the feature space for author gender identification problem and the features are normalized to treat all of them equally.

Then, we applied different classifiers using this feature space. We also applied several dimension reduc-

**Table 7**

The results in F1-score based on Word2vec

Classifier	Word2Vec only	PCA	LDA	RP	t-SNE	AE
k-NN	0.71	0.71	0.63	0.71	0.59	0.69
NB	0.66	0.66	0.70	0.66	0.60	0.62
SVM	0.76	<b>0.77</b>	0.69	0.76	0.66	0.71
DT	0.62	0.61	0.58	0.61	0.56	0.62
RF	0.73	0.72	0.59	0.73	0.58	0.72
SGD	0.63	0.62	0.67	0.62	0.52	0.70
LR	0.67	0.67	0.67	0.67	0.54	0.69
Boosting	0.72	0.71	0.70	0.72	0.67	0.61
Bagging	0.71	0.71	0.63	0.70	0.59	0.65
NC	0.61	0.60	0.68	0.60	0.53	0.68
MLP	0.73	0.73	0.67	0.74	0.62	0.70

**Table 8**

The results in F1-score based on Doc2vec

Classifier	Doc2Vec only	PCA	LDA	RP	t-SNE	AE
k-NN	0.68	0.69	0.68	0.68	0.59	0.78
NB	0.62	0.56	0.71	0.61	0.23	0.66
SVM	0.83	<b>0.84</b>	0.72	0.80	0.55	0.73
DT	0.59	0.65	0.64	0.58	0.57	0.65
RF	0.70	0.73	0.64	0.68	0.58	0.76
SGD	0.65	0.69	0.72	0.70	0.41	0.70
LR	0.71	0.71	0.72	0.70	0.18	0.71
Boosting	0.71	0.73	0.72	0.69	0.54	0.73
Bagging	0.69	0.69	0.68	0.68	0.59	0.78
NC	0.68	0.68	0.73	0.66	0.50	0.66
MLP	0.80	0.79	0.72	0.77	0.31	0.65

tion techniques such as PCA, LDA, NMF, RP, t-SNE and auto-encoder in order to capture the most discriminative features. Table 5 shows the scores of all classifiers based on SFs (SF-all). The best classification result produced by RF was the accuracy of 80% as suggested in [63]. Although SVM and NB are expected to perform well, their results were relatively low com-

pared to RF. While eliminating features improve the accuracy of classifier, the figure also reveals that discarding features did not lead to significant improvement of performance for most cases of the classifiers.

**Table 9**

The classification results of DNN in F1-score

Model	F1-score
RNN	0.53
Bi-RNN	0.62
LSTM	0.80
Bi-LSTM	0.80
GRU	0.58
Bi-GRU	0.68
CNN	<b>0.83</b>
C-LSTM	0.80
Hier. Atten. GRU	0.78
Hier. Atten. LSTM	0.77
MHA	0.82

**Table 10**

The best classification results

Feature	Classifier	Dim. Reduction	F1-score
SF	RF	-	0.80
BoW-tfidf	k-NN	-	0.87
Word2Vec	SVM	PCA	0.77
Doc2Vec	SVM	PCA	0.84
Word Embedding	CNN	-	0.83
fastText	fastText[30]	-	<b>0.87</b>

Experiments show that the performance of k-NN does not change significantly with number of features. Among all dimension reduction methods, only LDA improved the result of six classifiers.

Although NMF, RP, t-SNE and auto-encoder are promising methods and achieve great success for dimension reduction, they did not show significant difference in this experiment of the study. In some cases,

they even may cause loss of information, which has a negative effect on the accuracy of classifiers.

Experiments indicate that the most informative top twenty features are based on function-based, structure-based, word-based, and character-based features that are significant gender discriminators. While the average number of blank lines, words, characters and sentences per paragraph are the most discriminative of structure-based features, sport and health categories in function-based word features provide important information to identify the gender. Features that are specific for Turkish language such as emotion expressing words or symbols, and diacritics did not have positive effect on model. The list of SF's significances calculated by  $\chi^2$  test is given in Table 11.

The F1-scores of all classifiers across BoW representations by considering only the most frequent 15K is shown in Table 6. We chose tf-idf technique to represent weighted form of BoW of bi-grams. We eliminated LDA for BoW experiments due to the reason mentioned in Section 3.2. The other dimensional reduction techniques are applied to select the most informative features between classes.

Table 6 shows that k-NN and Bagging are the most successful algorithms with 87% F1-score. In text classification task, while some of the classifiers such as NB, LR, SVM are known to perform well for the BoW approach, some are known to be unsuccessful such as k-NN. However, k-NN shows unexpected performance with BoW-tfidf in this study. SGD and LR also performed well and have similar performance results. Although the NB algorithm has shown decent results on the author classification task [7], it produced 77% F1-score. Scores of DT and MLP algorithms also were worse than others.

Table 6 also shows the F1-scores of classifiers using dimension reduction algorithms. Despite some classifiers, a significant performance decrease was observed once the dimensions of BoW features were reduced. As in Table 5, NMF and auto-encoder did not contribute to the results.

In recent years, unsupervised word embedding models, particularly Word2Vec, have shown superior performances against embeddings utilizing distributed semantic information. However, these models require a considerable size of corpus to obtain accurate sense. Therefore, the embeddings pre-trained on

**Table 11**

The significance of SF

Order	Feature	Feature Description	p-value
1	Ft1	Tot. num. of characters	0.044
2	Ft55	Avg. num. of char. per par.	0.131
3	Ft49	Tot. num. lines	0.223
4	Ft57	Avg. num. words per sent.	0.310
5	Ft51	Tot. num. of paragraphs	0.402
6	Ft50	Tot. num. of sentences	0.470
7	Ft59	Avg. len. of non-blank line	0.567
8	Ft54	Avg. num. words per par.	0.592
9	Ft56	Avg. num. of sent. per par.	0.731
10	Ft58	Tot. num. of blank lines/ Tot. num. of lines	0.847
...			
107	Ft92	Geography Category	0.973
108	Ft64	Tot. num. of words longer than 6 char./W	0.977
109	Ft63	Tot. num. of short words (1-3 char.)/W	0.987
110	Ft72	Simpson's D measure	0.988
111	Ft93	Marine Category	0.990
112	Ft101	Grammar Category	0.992
113	Ft68	Hapax dislegomena/W	0.995
114	Ft100	Physic Category	0.996
115	Ft62	Tot. num. repeated word/W	0.999
116	Ft70	Sichel's S measure	1.000
117	Ft60	Absence/present of greeting words	1.000

a larger corpus are utilized for capturing vast semantic information on the language.

Table 7 shows the performance of the classifiers when used with embedding-based representations. The most successful classifier is SVM with 76% F1-score. Applying PCA with SVM increased the performance to 77%. In comparison with BoW results, Word2vec has definitely poor results.

Table 8 shows the scores of classifiers using Doc2vec. The results indicate that SVM again has better performance among other classifiers with 83% F1-score. Word embeddings effectively capture semantic relations between words and reflect word similarities through metric properties like distance and direction. However, for author gender identification purpose we need semantic relationships which span sentences and whole documents and not just words, as the gender classification is done on per document basis. When we compare the results of embedding models, Doc2vec with PCA achieves reasonably good performance and outperforms most of Word2vec and SF models.

We also utilized fastText as a simple and efficient classifier that is faster for training and evaluation than the other techniques. In this study, we utilized same parameters for fastText for the dataset and obtained 87% F1-score. The results show that fastText is probably the most suitable embedding model compared to other experimented models. It is also the most efficient method in terms of running time, being much faster than the other methods. As stated in [30], if the problem is a simple text classification, employing fastText might be simple and the right choice for evaluation.

Table 9 shows the F1-score of DNN architectures. CNN is the most successful model between the architectures with 83% F1-score. Chung et al. [18] experimented and shown that GRU outperformed LSTM on a suite of tasks, and pointed out that GRU can match LSTM's performance, and that its convergence speed sometimes outperforms LSTM. However, GRU underperformed all other architectures with accuracy of 58% in our experiment. Clearly, LSTM outperformed the more traditional RNNs on this task. Both hierarchical attention networks performed quite closely to each other.

Our experiment results demonstrate that DNN does not show better performance than traditional machine learning methods with regard to gender identification problem. Generally speaking, NB, SVM and LR are known to perform very well and have proven to be efficient and successful classification models for general such problems. However, k-NN classifier with BoW-tfidf has been surprisingly the most successful algorithm with 87% F1score in this study. Intuitively, while k-NN is expected to be less suitable for text mining problems, tree-based classifiers can be sometimes suitable. It is also shown that RF performs com-

paratively well in the approach based on stylometric features, with an accuracy of 80%. For the feature representation, many studies compare the traditional representations such as BoW and embedding based representations such as Word2vec. Even though BoW representation has the descriptive power to handle author gender problem, the drawback of the approach is the curse of the dimensionality. However, the results show that Word2vec did not demonstrate higher performance; this embedding transforms the representation of the words to a space where structure of semantic word relations are approximated by the metric distance on the embedded space. Some semantic analysis studies have shown that embedding representations are very effective and useful for purposes related to word semantics [41, 23, 56, 45], and fastText embedding have shown great performance in author gender identification with 87% F1-score. While it is competitive with BoW-tfidf in terms of resulting classification accuracy, it is also the simple and efficient one based on computational complexity. Although the performance of the DNN depends on hyper-parameters and configuration of the topology such as number of dimensions, the size of dataset is the driving factor to obtain good results. As the size of corpus increases, naturally the results are expected to improve. In this study, we have constructed our own word/document embeddings using Word2vec, fastText and Doc2vec based on these Turkish corpora with size of 13.7M tokens instead of using a pre-trained one. Larger embeddings based on different very large datasets different from news articles and employing different benchmarks will be part of further study. Also, it is important to quantify the impact of using different Turkish embeddings (pre-trained embeddings) on the obtained performance. In this study, the model we propose aims to be as generic as possible for the whole Turkish Language, and not domain specific, and it is also adaptable to any other language. Although we selected language-independent stylometric features, the significance of the features may differ for languages. Since Turkish is an agglutinative language that is morphologically rich, the effects of word-based features would be more compared to other languages. Intuitively, a similar effect may present for German because of the high frequency of compound words usage. Further empirical studies conducted in multi-languages would reveal the stylometric differences between languages.

## 5. Conclusion

In this study, two learning approaches are proposed to address author gender identification for Turkish articles. First, two different feature representation approaches are on par with ML classifiers in terms of F1-scores. While SF features represented with BoW serve as a fixed length vector representation, Word2vec, fastText and Doc2vec are used as embedding based distributed vector representations. Secondly, we built and trained implementations of well-known DNN architectures and evaluated the associated performances.

The results show that RF with SF approach has 80% F1-score among other classifiers and the k-NN with BoW-tfidf has the highest F1-score with 87%. CNN achieved the highest performance between DNN architectures with 83%. SVM with Doc2vec have achieved 84% with

comparable performance. In addition, fastText is the most successful embedding structure with 87% F1-score compared to other representations under same training setup; it is also simple in implementation and efficient in terms of computational complexity.

In conclusion, this study is the first attempt at using SF on the author gender identification task for the Turkish language to the best of our knowledge. A comparative analysis was conducted on utilizing SF and other text representations. Since the presented set of SF and embedding structures are language-independent, further experiments can be carried out on different languages. Therefore, this study raises the question of how the significance of stylometric features changes depending on the language for future studies.

## References

1. Abbasi, A., Chen, H. Applying Authorship Analysis to Arabic Web Content. In *International Conference on Intelligence and Security Informatics*, Springer, 2005, 183-197. [https://doi.org/10.1007/11427995\\_15](https://doi.org/10.1007/11427995_15)
2. Alsmearat, K., Al-Ayyoub, M., Al-Shalabi, R. An Extensive Study of the Bag-of-Words Approach for Gender Identification of Arabic Articles. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2014, 601-608. <https://doi.org/10.1109/AICCSA.2014.7073254>
3. Alsmearat, K., Al-Ayyoub, M., Al-Shalabi, R., Kanaan G. Author Gender Identification from Arabic Text. *Journal of Information Security and Applications*, 2017, 35, 85-95. <https://doi.org/10.1016/j.jisa.2017.06.003>
4. Alsmearat, K., Shehab, M., Al-Ayyoub, M., Al-Shalabi, R., Kanaan, G. Emotion Analysis of Arabic Articles and Its Impact on Identifying the Author's Gender. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, IEEE, 2015, 1-6. <https://doi.org/10.1109/AICCSA.2015.7507196>
5. AlSukhni, E., Alequr, Q. Investigating the Use of Machine Learning Algorithms in Detecting Gender of the Arabic Tweet Author. *International Journal of Advanced Computer Science and Applications*, 7(7), 2016, 319-328. <https://doi.org/10.14569/IJACSA.2016.070746>
6. Altamimi, M., Teahan, W. J. Gender and Authorship Categorisation of Arabic Text from Twitter Using ppm. *International Journal of Computer Science and Information Technology*, 9, 2017, 131-140. <https://doi.org/10.5121/ijcsit.2017.9212>
7. Amasyalı, M. F., Diri, B. Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. In *International Conference on Application of Natural Language to Information Systems*, Springer, 2006, 221-226. [https://doi.org/10.1007/11765448\\_22](https://doi.org/10.1007/11765448_22)
8. Argamon, S., Koppel, M., Pennebaker, J. W., Schler, J. Mining the Blogosphere: Age, Gender and the Varieties of Self-expression. *First Monday*, 2007, 12(9). <https://doi.org/10.5210/fm.v12i9.2003>
9. Bayot, R. K., Gonçalves, T. Author Profiling Using SVMs and Word Embedding Averages. In *CLEF (Working Notes)*, 2016, 815-823. <https://doi.org/10.1109/SKI-MA.2016.7916251>
10. Bengio, Y., Ducharme, R., Vincent P., Jauvin, C. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 2003, 3, 1137-1155.
11. Bengio, Y., Simard, P., Frasconi, P., et al. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 1994, 5(2), 157-166. <https://doi.org/10.1109/72.279181>

12. Bojanowski, P., Grave, E., Joulin, A., Mikolov T. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 2017, 5, 135-146. [https://doi.org/10.1162/tacl\\_a\\_00051](https://doi.org/10.1162/tacl_a_00051)
13. Bsir, B., Zrigui, M. Bidirectional lstm for Author Gender Identification. In *International Conference on Computational Collective Intelligence*, Springer, 2018, 393-402. [https://doi.org/10.1007/978-3-319-98443-8\\_36](https://doi.org/10.1007/978-3-319-98443-8_36)
14. Burger, J. D., Henderson, J., Kim, G., Zarrella, G. Discriminating Gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2011, 1301-1309.
15. Cheng, N., Chandramouli, R., Subbalakshmi, K. Author Gender Identification from Text. *Digital Investigation*, 2011, 8(1), 78-88. <https://doi.org/10.1016/j.diin.2011.04.002>
16. Cheng, N., Chen, X., Chandramouli, R., Subbalakshmi, K. Gender Identification from e-mails. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, IEEE, 2009, 154-158.
17. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning Phrase Representations Using rnn Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*, 2014. <https://doi.org/10.3115/v1/D14-1179>
18. Chung J., Gulcehre C., Cho K. and Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
19. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. *Natural Language Processing (almost) from Scratch*. *Journal of Machine Learning Research*, 2011, 12, 2493-2537.
20. Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T., Hu, W. Author Gender Prediction in an email Stream Using Neural Networks. *Journal of Intelligent Learning Systems and Applications*, 2012, 4(03). <https://doi.org/10.4236/jilsa.2012.43017>
21. Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T., Hu, W. Gender Identification on Twitter Using the Modified Balanced Winnow. *Communications and Network*, 2012, 4(3), 189-195. <https://doi.org/10.4236/cn.2012.43023>
22. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. Bert Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.
23. Fu, R., Guo, J., Qin, B., Che, W., Wang, H., Liu T. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, 1199-1209. <https://doi.org/10.3115/v1/P14-1113>
24. Graves, A., Jaitly, N. Towards End-to-End Speech Recognition with Recurrent Neural Networks. In *International Conference on Machine Learning*, 2014, 1764-1772.
25. Graves, A., Jaitly, N., Mohamed, A.-r. Hybrid Speech Recognition with Deep Bidirectional LSTM. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, IEEE, 2013, 273-278. <https://doi.org/10.1109/ASRU.2013.6707742>
26. Graves, A., Mohamed, A.-r. and Hinton G. Speech Recognition with Deep Recurrent Neural Networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, 6645-6649. <https://doi.org/10.1109/ICASSP.2013.6638947>
27. Guo, Q., Qiu, X., Liu, P., Xue, X., Zhang, Z. Multi-scale Self-attention for Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, 5, 7847-7854. <https://doi.org/10.1609/aaai.v34i05.6290>
28. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-term Dependencies, 2001.
29. Hochreiter, S., Schmidhuber, J. Long Short-term Memory. *Neural Computation*, 1997, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
30. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*, 2016. <https://doi.org/10.18653/v1/E17-2068>
31. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1408.5882*, 2014. <https://doi.org/10.3115/v1/D14-1181>
32. Kodyan, D., Hardegger, F., Neuhaus, S., Cieliebak, M. Author Profiling with Bidirectional rnns Using Attention with grus: Notebook for pan at clef 2017. In *CLEF 2017 Evaluation Labs and Workshop-Working Notes Papers*, Dublin, Ireland, 11-14 September 2017. RWTH Aachen, 2017.
33. Koppel, M., Argamon, S., Shimoni, A. R. Automatically Categorizing Written Texts by Author Gender. *Literary and Linguistic Computing*, 2002, 17(4), 401-412. <https://doi.org/10.1093/lc/17.4.401>
34. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. E., Brown, D. E. Text Classification Algorithms: A Survey. *Information*, 2019, 10(4). <https://doi.org/10.3390/info10040150>

35. Lai, S., Xu, L., Liu, K., Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Twenty-ninth AAAI Conference on Artificial Intelligence, 2015. <https://doi.org/10.1609/aaai.v29i1.9513>
36. Le, Q., Mikolov, T. Distributed Representations of Sentences and Documents. In International Conference on Machine Learning, 2014, 1188-1196.
37. Liu, W., Ruths, D. What's in a Name? Using First Names as Features for Gender Inference in Twitter. In 2013 AAAI Spring Symposium Series, 2013.
38. Markov, I., Gómez-Adorno, H., Posadas-Durán, J.-P., Sidorov, G., Gelbukh, A. Author Profiling with doc2vec Neural Network-based Document Embeddings. In Mexican International Conference on Artificial Intelligence, Springer, 2016, 117-131. [https://doi.org/10.1007/978-3-319-62428-0\\_9](https://doi.org/10.1007/978-3-319-62428-0_9)
39. Marquardt, J., Farnadi, G., Vasudevan, G., Moens, M.-F., Davalos, S., Teredesai, A., De Cock, M. Age and Gender Identification in Social Media. Proceedings of CLEF 2014 Evaluation Labs, 2014, 1180, 1129-1136.
40. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Advances in Neural Information Processing Systems, 2013, 3111-3119.
41. Mikolov, T., Yih, W.-t., Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, 746-751.
42. Modak, S., Mondal, A. C. A Comparative Study of Classifiers' Performance for Gender Classification. International Journal of Innovative Research in Computer and Communication Engineering, 2014, 2(5), 4214-4222.
43. Mukherjee, A., Liu B. Improving Gender Classification of Blog Authors. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2010, 207-217.
44. Nowson, S., Oberlander, J. The Identity of Bloggers: Openness and Gender in Personal Weblogs. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, Palo Alto, CA, 2006, 163-167.
45. Rei, M., Briscoe T. Looking for Hyponyms in Vector Space. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning, 2014, 68-77. <https://doi.org/10.3115/v1/W14-1608>
46. Sak, H., Güngör, T., Saraçlar, M. Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus. In International Conference on Natural Language Processing, Springer, 2008, 417-427. [https://doi.org/10.1007/978-3-540-85287-2\\_40](https://doi.org/10.1007/978-3-540-85287-2_40)
47. Sap, M., Park, G., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., Ungar, L., Schwartz, H. A. Developing Age and Gender Predictive Lexica Over Social Media. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, 1146-1151. <https://doi.org/10.3115/v1/D14-1121>
48. Sboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., Moloshnikov, I. Machine Learning Models of Text Categorization by Author Gender Using Topic-Independent Features. Procedia Computer Science, 2016, 101, 135-142. <https://doi.org/10.1016/j.procs.2016.11.017>
49. Sboev, A., Litvinova, T., Voronina, I., Gudovskikh, D., Rybka, R. Deep Learning Network Models to Categorize Texts According to Author's Gender and to Identify Text Sentiment. In 2016 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2016, 1101-1106. <https://doi.org/10.1109/CSCI.2016.0210>
50. Schler, J., Koppel, M., Argamon, S., Pennebaker, J. W. Effects of Age and Gender on Blogging. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006, 199-205.
51. Schuster, M., Paliwal, K. K. Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing, 1997, 45(11), 2673-2681. <https://doi.org/10.1109/78.650093>
52. Semeniuta, S., Severyn, A., Barth E. Recurrent Dropout Without Memory Loss. arXiv preprint arXiv:1603.05118, 2016.
53. Severyn, A., Moschitti, A. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, 959-962. <https://doi.org/10.1145/2766462.2767830>
54. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., Potts, C. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, 1631-1642.
55. Talebi, M., Köse, C. Identifying Gender, Age and Education Level by Analyzing Comments on Facebook. In 2013 21st Signal Processing and Communications Applications Conference (SIU), IEEE, 2013, 1-4. <https://doi.org/10.1109/SIU.2013.6531599>
56. Tan, L., Gupta, R., van Genabith, J. Usaar-wlv: Hypernym Generation with Deep Neural Nets. In Proceed-



- ings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, 932-937. <https://doi.org/10.18653/v1/S15-2155>
57. Tjandra, A., Sakti, S., Manurung, R., Adriani, M., Nakamura, S. Gated Recurrent Neural Tensor Network. In 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, 448-455. <https://doi.org/10.1109/IJCNN.2016.7727233>
  58. Turian, J., Ratinov, L., Bengio, Y. Word Representations: A Simple and General Method for Semi-Supervised Learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, 384-394.
  59. Ugheoke, T. O., Saskatchewan, R. Detecting the Gender of a Tweet Sender. A Project Report Submitted to the Department of Computer Science in Partial Fulfilment of the Requirements for the Degree of Master of Science in Computer Science, University of Regina, Saskatchewan, 2014.
  60. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N, Kaiser, Ł., Polosukhin, I. Attention is All You Need. In Advances in Neural Information Processing Systems, 2017, 5998-6008.
  61. Volkova, S., Wilson, T., Yarowsky, D. Exploring Demographic Language Variations to Improve Multilingual Sentiment Analysis in Social Media. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, 1815-1827.
  62. Wang, X., Liu, Y., Chengjie, S., Wang, B., Wang, X. Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, 1343-1353. <https://doi.org/10.3115/v1/P15-1130>
  63. Xu, B., Guo, X., Ye, Y., Cheng, J. An Improved Random Forest Classifier for Text Categorization. JCP, 2012, 7(12), 2913-2920. <https://doi.org/10.4304/jcp.7.12.2913-2920>
  64. Yan, X., Yan, L. Gender Classification of Weblog Authors. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, Palo Alto, CA, 2006, 228-230.
  65. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, 1480-1489. <https://doi.org/10.18653/v1/N16-1174>
  66. Yao, K., Cohn, T., Vylomova, K., Duh, K., Dyer, C. Depth-gated LSTM. arXiv preprint arXiv:1508.03790, 2015.
  67. Yildiz, T. A Comparative Study of Author Gender Identification. Turkish Journal of Electrical Engineering and Computer Science, 2019, 27(2), 1052-1064. <https://doi.org/10.3906/elk-1806-185>
  68. Yin, W., Kann, K., Yu, M., Schütze, H. Comparative Study of cnn and rnn for Natural Language Processing. arXiv preprint arXiv:1702.01923, 2017.
  69. Young, T., Hazarika, D., Poria, S., Cambria E. Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine, 13(3), 2018, 55-75. <https://doi.org/10.1109/MCI.2018.2840738>
  70. Zhang, Y., Wallace, B. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1510.03820, 2015.
  71. Zhou, C., Sun, C., Liu, Z., Lau, F. A C-LSTM Neural Network for Text Classification. arXiv preprint arXiv:1511.08630, 2015.
  72. Zhou, W., Wang, H., Sun, H., Sun, T. A Method of Short Text Representation Based on the Feature Probability Embedded Vector. Sensors, 2019, 19(17), 3728. <https://doi.org/10.3390/s19173728>

