Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1052 – 1064

doi:10.3906/elk-1806-185

A comparative study of author gender identification

Tuğba YILDIZ*
Department of Computer Engineering, Faculty of Engineering and Natural Science, İstanbul Bilgi University, İstanbul, Turkey

Received: 25.06.2018 • **Accepted/Published Online:** 10.12.2018 • **Final Version:** 22.03.2019

Abstract: In recent years, author gender identification has gained considerable attention in the fields of information retrieval and computational linguistics. In this paper, we employ and evaluate different learning approaches based on machine learning (ML) and neural network language models to address the problem of author gender identification. First, several ML classifiers are applied to the features obtained by bag-of-words. Secondly, datasets are represented by a low-dimensional real-valued vector using Word2vec, GloVe, and Doc2vec, which are on par with ML classifiers in terms of accuracy. Lastly, neural networks architectures, the convolution neural network and recurrent neural network, are trained and their associated performances are assessed. A variety of experiments are successfully conducted. Different issues, such as the effects of the number of dimensions, training architecture type, and corpus size, are considered. The main contribution of the study is to identify author gender by applying word embeddings and deep learning architectures to the Turkish language.

Key words: Author gender identification, convolution neural network, recurrent neural network, Word2vec, Doc2vec

1. Introduction

The availability of large amounts of texts obtained through the Internet and the anonymity of the texts have revealed the potential of authorship analysis, which deals with the task of authorship attribution where knowing the author of a document, or with the task of authorship profiling, where the authors' personality type, age, and gender are determined. Authorship analysis could be considered as a text classification problem that maps documents to a certain category from a predefined list. Author gender identification, which is a subproblem of the authorship profiling problem, aims at determining the gender of an author of a given text.

In recent years, author gender identification has gained importance in various commercial applications including e-mail forgery, online communities, security, forensics, trading, and marketing. People avoid providing their real identity information on social media platforms, which leads to an anonymity problem. Also, companies often need the demographic information of people for direct marketing, risk management, fraud detection, etc. Therefore, the question "Can we identify author gender for a given text?" is very relevant for practical applications.

Various studies and scientific events are devoted to author gender identification. For example, PAN is an important series of scientific events, where tasks on digital text forensics are assigned. One of these tasks is predicting authors' demographics from their writings.

A key point in the author gender identification problem is the representation of features. One popular

*Correspondence: tugba.yildiz@bilgi.edu.tr

representation is the writing style of an author, which can be characterized through stylometric features (SFs). 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, word-based, syntactic-based, structure-based, and function words. All these features can be automatically extracted and the classification models based on these features can classify the author gender of a candidate text. Another widely used feature extraction is based on bag-of-words (BoW). The same classification process can be evaluated by BoW with different weighting schemes or other techniques such as character/word n-gram, tf-idf, etc.

Recently, neural network language models (NNLM) have been employed to learn distributed representations (word embeddings), which have been effectively applied to the several problems in natural language processing (NLP). Many training approaches have been proposed by [1–3] and [4] showed that word embeddings are good at capturing syntactic and semantic regularities using the vector offsets between word pairs sharing a particular relation. Later, different architectures were presented in [5], namely continuous bag-of-words (CBoW) and the skip-gram (SG) for training word embeddings efficiently to minimize computational complexity and maximize accuracy. In [4], Word2vec was presented as a tool providing an efficient implementation of CBoW and SG architectures for computing vector representations of words. In [6], GloVe, an unsupervised learning algorithm, was described for obtaining word vector representations based on matrix factorization and a new global log-bilinear regression model. Similar to the Word2vec approach, Doc2vec or Paragraph vectors, described in [7], perform an unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs, or entire documents. However, all these techniques are not fully NNLM-based; they only turn text into a numerical form that neural networks architectures can understand. At that point, the convolutional neural network (CNN) and recurrent neural network (RNN), which are the two main types of complicated neural network architectures, are also utilized for addressing various NLP tasks [8].

In this study, models based on machine learning (ML) and deep learning architectures are proposed to address the author gender identification problem for Turkish articles. First, eight different classifiers are applied to features that are obtained by BoW. Then classifiers are built considering word/document embeddings, specifically Word2vec, GloVe, and Doc2vec. Lastly, a CNN and RNN are trained and their performances are assessed. Various experiments are conducted and all models are compared with respect to several criteria. It is observed that the proposed model gives promising results for the Turkish language. This study is considered to be the first important attempt that utilizes particular embeddings and complicated neural networks architectures for the identification of author gender in the Turkish language.

2. Related work

In recent years, various studies have focused on author gender identification automatically. In [9], combinations of simple lexical and syntactic features were used to infer the author gender of the British National Corpus (BNC) with 80% accuracy. One conclusion of this work was that a winnow-like algorithm outperforms less subtle techniques such as decision tree (DT) and naive Bayes (NB). Similarly, in [10], both the balanced and modified balanced winnow (MBW) algorithms were used to predict gender for the Enron e-mail dataset using a set of SFs and word count features. Here, the MBW algorithm outperformed the balanced winnow algorithm and showed 88% accuracy for SF and 95% for word-based features. In [11], the MBW algorithm was also exploited for identifying a user's gender on Twitter with 82.48% accuracy. The accuracy was increased to 98.51% using feature selection methods.

Another study on determining gender on Twitter was proposed in [12]. The authors utilized a number of text-based features and several different classifiers, including balanced winnow, NB, and support vector machines (SVM). Some other studies presented gender classification of blog authors using features such as part-of-speech tags, content words, and feature selection methods [13–15].

In [16], a model was proposed for identifying author gender via e-mails. The authors also carried out a training of DT and SVM using word-based and function words features. As a result, the SVM method outperformed the DT method. They showed that the roles of the word-based features and function words are important for gender identification. In [17], 545 features were identified and classified into five sets: character-based, word-based, syntactic, structure-based, and function words. Three classifiers (SVM, Bayesian logical regression, AdaBoost) were designed and SVM outperformed the others with 76.75% and 82.23% accuracy for two different datasets.

In [18], the BoW approach was followed by considering a set of feature selection/reduction techniques (including stemming) and applying a list of classifiers. The best classifier was found to be SGD with 94.1% accuracy. In an extended work [19], the authors also extracted features based on sentiments and emotions (BoW-EF) in addition to BoW features. They showed that BoW-EF features did not improve the accuracy. In [20], the authors extended [18, 19] by considering SFs. They repeated all the experiments based on SFs and the BoW approach for the Arabic language. They stated that the BoW approach is expensive and less accurate than the SF approach. While the accuracy of the SF approach was 80.4%, the BoW approach showed 73.9% accuracy.

Traditional feature representations describe word meanings as points in high-dimensional space. However, high dimensionality and sparsity can be ambiguous and insufficient. Thus, recently NNLMs are exploiting their ability to learn distributed representation. Word embeddings are a modern approach for representing text. These dense and low-dimensional real-valued vectors have been effectively applied to the semantic and syntactic problems. In [21], the authors used features that were obtained from averages of word embeddings, specifically Word2vec, and trained using the SVM classifier to address author gender and age classification problems. Using the PAN 2016 dataset, they achieved 44.8% and 68.2% accuracies for age and gender classification for English, respectively. In [22], they used Doc2vec to train a logistic regression (LR) classifier on the PAN author profiling 2014–2016 corpora. They compared the document embedding features with traditional features.

In [23, 24], two different approaches were proposed for automatic text classification according to author gender in the Russian language. In a first approach, different ML algorithms were used. A CNN was utilized as a second approach, with accuracy of 86%.

In Turkish, [25] proposed term-based and style-based approaches to predict the attributes of authors such as age and gender for chat messages. They obtained accuracies of 82.2% in prediction of gender using SVM with a term-based feature set. In [26], they considered the author gender identification problem using discriminant analysis and analyzed the change of frequent word usage with gender. The authors of [27] presented an n-gram model for identifying the gender of authors. C4.5, NB, SVM, and random forest were used as classifiers and correlation-based feature selection (CFS) were applied to obtain the feature subset. The dataset included 4 female and 14 male authors. SVM outperformed all classifiers with 96.3% accuracy.

3. Methodology

3.1. Data

The dataset was manually collected from several Turkish news websites. We tested our models on a dataset with 2.76K articles evenly distributed across genders. The dataset consists of ten articles for each male and female author. The distributions of articles across the authors and genders are the same. The summary statistics of datasets are given in Table 1. It includes the number of authors, the number of total articles, and some basic statistics on characters, words, and sentences. The collected articles were preprocessed in several steps to extract features using text processing libraries of the Natural Language Toolkit (NLTK).[1] We removed punctuation and special characters, replaced nonletters with white space, converted capitals to lowercase, eliminated stop words, and did not apply stemming for Turkish.

Table 1. Summary of three datasets.

Features	Male	Female
#ofauthors	138	138
#ofarticles	1380	1380
avg#ofsent/article	35.5	36.5
avg#ofwords/article	579.4	531.3
avg#ofchar/article	4495.4	4104.2
avg#ofchar/sent	126.3	112.1
avg#ofchar/word	7.7	7.7
avg#ofwords/sent	16.2	14.5

3.2. Models

In this study, we applied two different models for the problem of author gender identification. For the first model, we utilized eight different ML classifiers: k-NN (k-nearest neighbors), naive Bayes (NB), logistic regression (LR), support vector machine (SVM), stochastic gradient descent (SGD), decision tree (DT), random forest (RF), and multilayer perceptron (MLP). For feature representations, we exploited several variants of BoW features. In addition, the classifiers were trained with the representation obtained through embeddings: Word2vec, GloVe, and Doc2vec. Besides that, two main types of neural networks architectures, CNN and RNN, are employed for training.

3.2.1. Machine learning methods

We conducted experiments with eight different ML classifiers using different representation schemes. The traditional feature vector representation is BoW or bag of n-grams representation, which represents a text as a collection of words or n-grams by completely ignoring the position of the words in the document. This traditional approach has been widely used in several tasks such as document classification, document similarity, etc. In this study, we extracted both uni- and bigrams (BoW-uni/bi) of words in addition to character n-gram representation (BoW-char). We also utilized tf-idf as the weighting schema for vector representation. We also applied BoW-tfidf to observe whether it contributes to improving the accuracy of the model or not.

[1] http://www.nltk.org/

In the BoW approach, we built a vocabulary considering only the most frequent 5K and 10K features. We tested the size of vocabulary in the range of 5K–15K; however, we did not see any significant difference in the performance for the n-fold (n = 10) cross-validation. We also used modules that implement feature selection algorithms χ^2 and mutual information using Python scikit-learn.[2]

Traditional feature representations generate high-dimensional feature vectors and lead to sparsity problems. High dimensionality and sparsity can be ambiguous and insufficient, and ignore contextual information and word ordering. For this reason, recently distributed vector representation (word embeddings) have been more effectively used in NLP tasks. In this study, word and document embeddings obtained by Word2vec, GloVe, and Doc2vec are also utilized to generate word and document vectors. The experiments were conducted with Gensim,[3] NLTK libraries, and GloVe.[4]

Most studies experimentally select embedding size (K) depending on the problem and the approach. To test the influence of vector size, we gradually vary the dimension size K from 100 to 500 by steps of 100. Accordingly, we decided to set the K value to 300. For window size selection, some studies [28] showed that smaller contextual windows generally give better precision. After some experiments, we set the window size to 10. For word vectors, we run the experiments using both SG and CBoW training algorithms with negative sampling (NS) and hierarchical softmax (HS). We observed that there are no significant differences between the SG and CBoW training algorithms for the classification model.

The document embeddings are based on paragraph-vector distributed memory (PV-DM) and a distributed BoW version of paragraph vector (PV-DBOW). PV-DM averages or concatenates the paragraph vector into the context window for all windows of the document. PV-DBOW creates paragraph vectors by training to predict words within a window of the paragraph. While PV-DM incorporates word order in training, PV-DBOW ignores word order. Thus, PV-DBOW is very similar to the CBoW method for training word vectors.

3.2.2. Neural network architectures

Neural networks with pretrained word embeddings have been shown to be effective for NLP tasks [2, 29–32]. In this study, we used two popular neural network architectures, namely CNN and RNN. In [29], the authors experimented with several variants of the CNN model, such as CNN-static, CNN-rand, etc. CNN-static utilizes pretrained vectors from Word2vec. The model learns the parameters by taking word embeddings as input. In the CNN-rand model, all words are randomly initialized and then modified during training. In this study, we trained a CNN model with word embeddings obtained by Word2vec.

An unbiased model, the CNN utilizes layers involving filters that are applied to local features [33]. In the CNN architecture, let x_i be the k-dimensional word vector corresponding to the ith word in the sentence, and n represents the sentence length.

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n. \tag{1}$$

In Eq. (1), $\oplus$ is the concatenation operator. Let $x_{i:i+j}$ refer to the concatenation of all words $x_i, x_{i+1}, ..., x_{i+j}$. As expressed in Eq. (2), a filter w is applied to a window of h words to produce a new feature c_i:

$$c_i = f(w.x_{i:i+h-1} + b), \tag{2}$$

[2]http://scikit-learn.org/stable/
[3]https://radimrehurek.com/gensim/
[4]https://github.com/stanfordnlp/GloVe

where b is a bias term and f is a nonlinear function such as the hyperbolic tangent. The filter w is applied to each possible window of words in the sentence to produce a feature map as expressed in Eq. (3):

$$c = [c_1, c_2, ..., c_{n-h+1}]. \tag{3}$$

Then a max pooling operation [2] over the feature map is applied and the maximum value $\hat{c} = max\{c\}$ is taken to find the most important feature for each feature map.

The RNN is one of the possible neural network architectures for modeling sequential and temporal dependencies [34]. In this architecture, given input sequence $x = (x_1, ...x_t)$, the hidden vector sequence $h = (h_1, ...h_t)$ and output vector sequence $y = (y_1, ...y_t)$ are calculated according to Eqs. (5)–(7).

$$h_t = f(W_h[x_t, h_{t-1}] + b_h), \tag{5}$$

$$h_t = f(x_t W_{xh} + h_{t-1} W_{hh} + b_h), \tag{6}$$

$$y_t = g(h_t W_{hy} + b_y), \tag{7}$$

where x_t is the input at time step t; h_t and h_{t-1} are hidden vectors that encode the current and previous states of the network; W_h is the 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 is a vector of bias terms; b_h represents bias vectors for the hidden layer; and b_y is the bias vector for the output layer. $f(.)$ and $g(.)$ are nonlinear activation functions such as sigmoid or tanh.

For the model hyperparameters, we utilized embedding size of 300, unit size of 32, filter size of 64, and dropout rate of 0.5. For training parameters, we tested batch sizes between 32 and 64. We employed callbacks, EarlyStopping, which stops training when a monitored quantity has stopped improving on development sets. The training generally stops after 9–10 epochs. Sentence lengths are automatically calculated for the dataset. We randomly selected 10% of the training data as the development set. For Word2vec parameters, we used the minimum word count of 1 and window size of 10. The experiments were conducted with Keras,[5] which is a high-level neural networks API, coded in Python.

4. Preliminary analysis

To understand the word usage across the genders, we mapped the words to a 2D semantic space. Figure 1 provides important pieces of information where the x and y axes point to how often a word appears in female contexts and male contexts, respectively. We used logarithmic document frequency (log-df) to fairly scatter the points and analyzed them. This figure further shows general document frequency of the word as indicated by the diameters of the circles. It can be seen that although many common words are shared across genders, some informative gender-dependent terms are found to scatter in the corresponding parts of the graph. The upper left region, as male context, and the lower right region, as female context, include the informative words. Upon closer inspection, even though the numbers of these distant and informative words are very low, they show the traditional gender stereotypes. In the male context as shown in Figure 2a, the words are mostly related to politics, sports, and business. On the other hand, in the female context shown in Figure 2b, words include the topics of kitchen and family. This kind of discussion is beyond the scope of this paper.

[5] https://keras.io/

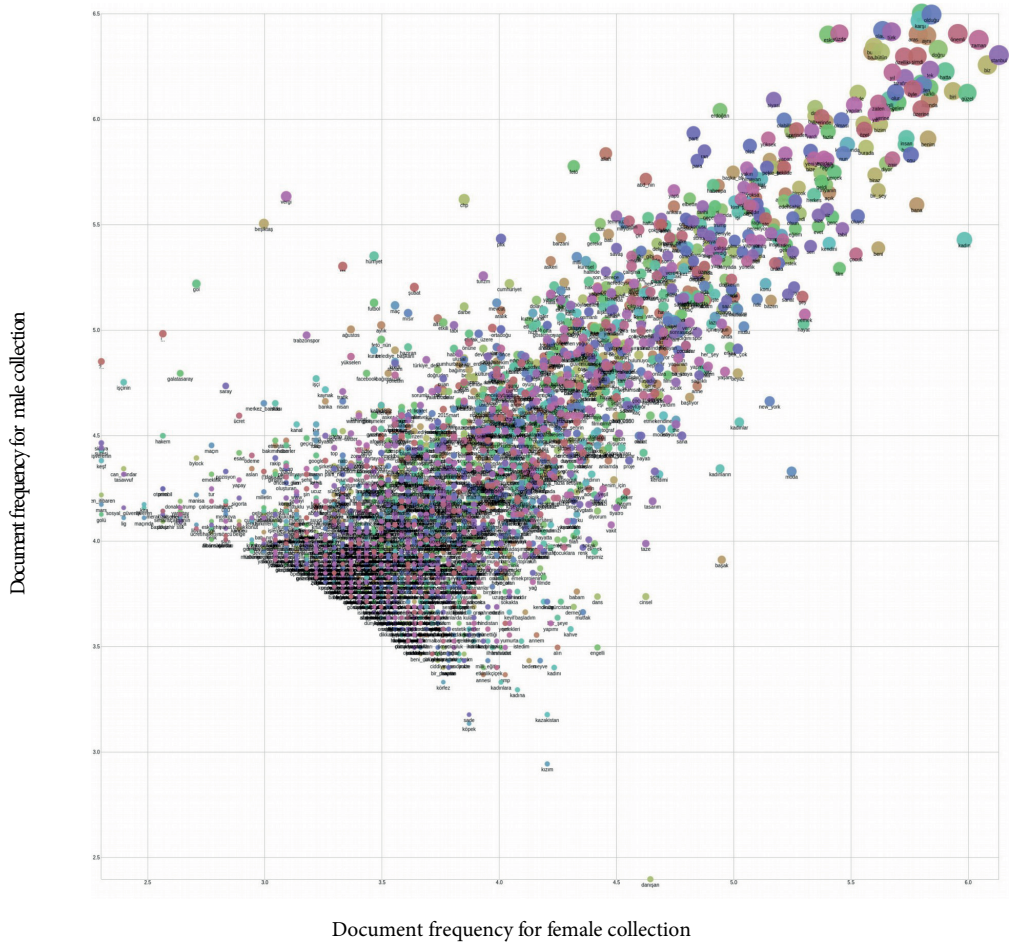

Figure 1. Document frequency across genders.

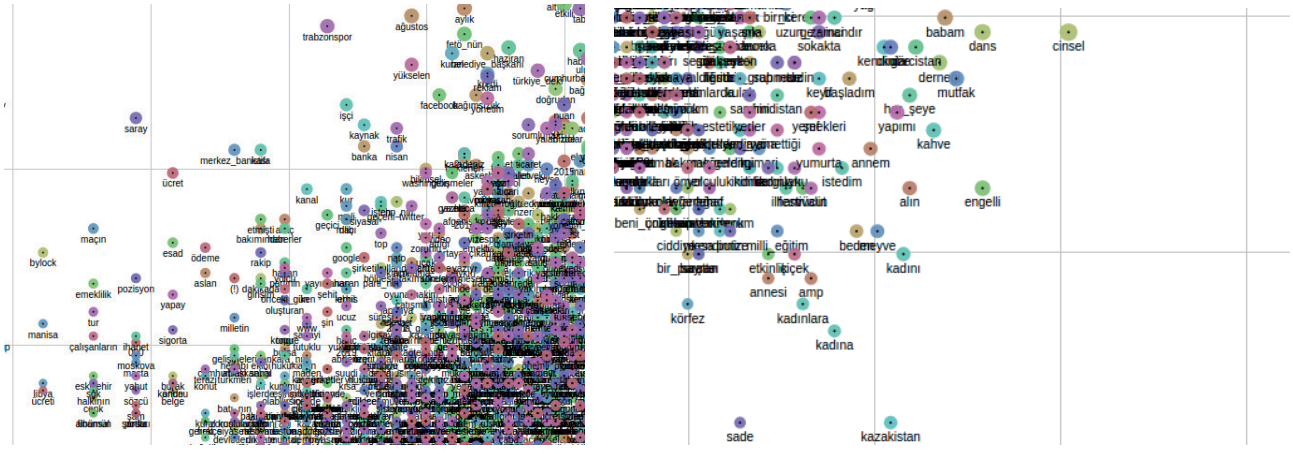

Figure 2. A set of two subfigures describes: (a) projection from male context; (b) projection from female context.

Another observation is the analysis of the length of the articles. Figure 3 shows the box-plot representation of the article length across genders. Interestingly, the average length of male articles is higher than the average length of female articles.

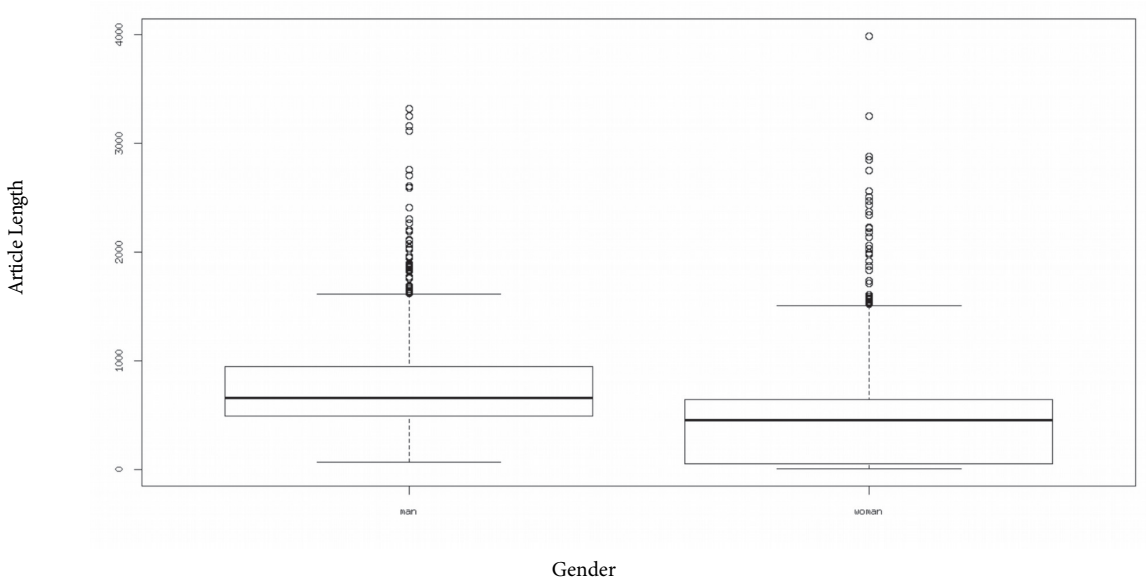

Figure 3. Article length across genders.

5. Experimental results

We measured the performances of the models in terms of accuracy, F1, recall, and precision. The F1-score is only considered as our main metric. Table 2 shows all the scores of all ML algorithms across different BoW representations by considering only the most frequent 5K and 10K features. First we compared the results of BoW-char, BoW-uni/bi, and BoW-tfidf. However, experiments indicated that BoW-char had low performance compared to all other variants of BoW. For this reason, we eliminated BoW-char and continued the experiments with BoW-uni/bi and BoW-tfidf. BoW-tfidf is also calculated using the same range of n-values as BoW-uni/bi. As expected, the comparison indicates that the results of BoW-tfidf outperform BoW-uni/bi because of the composite weight employed for each term in each document. We also applied feature selection using mutual information and χ^2 to select the most informative features. While the former does not improve the performance, the latter contributes to improving performance by selecting the 5K and 10K features with the highest values for χ^2.

While the k-NN classifier with BoW-tfidf is the most successful algorithm with an F1-score of 81% for 5K feature size, it has 82% for 10K feature size. SVM performed well and became the second most successful classifier among other classifiers. Table 2 shows also the F1-scores of classifiers using the χ^2 feature selection criterion for BoW-tfidf. While the results indicate that the performance of all classifiers increases, the performance of k-NN is dramatically decreased and shows slightly lower success ratio when the feature selection algorithm is applied. Feature selection eliminates features with low discrimination power, leading to improved performance so that the accuracy of some classifiers such as k-NN is decreased. The SGD classifier outperformed all other classifiers with 91% for 5K and 93% for 10K feature size. While NB is already one of the best classifiers for the text classification problem, it produced the worst results because of its reliance on feature independence.

Table 2. The F1-scores of classifiers based on BoW.

Dim: 5K	uni/bi	tfidf	χ^2
k-NN	65	**81**	68
NB	71	70	83
LR	74	76	81
SVM	73	77	83
SGD	72	76	**91**
DT	63	65	65
RF	59	61	63
MLP	75	76	71
Dim: 10K	uni/bi	tfidf	χ^2
k-NN	65	**82**	67
NB	72	71	83
LR	75	77	83
SVM	73	79	87
SGD	73	78	**93**
DT	65	64	65
RF	60	62	62
MLP	75	66	64

Table 3 shows the model performance of the classifiers for word embedding representations using Word2vec and GloVe. The results indicate that Word2vec is more successful than GloVe in overall performance. Among classifiers, k-NN outperforms other ML algorithms again. There is no significant difference between MLP, SVM, and LR in some cases.

Table 3 also gives the scores of classifiers using Doc2vec. The results indicate that MLP has better performance among other classifiers with 84% F1-score. Word embeddings effectively capture semantic relations between words and calculate word similarities. However, we need relationships between sentences and documents and not just words for the classification based on documents. When we compare the results of embedding models, Doc2vec achieves reasonably good performance and outperforms GloVe and most of the Word2vec models.

Table 3. The F1-scores of classifiers based on Word2vec, GloVe, and Doc2vec.

Classifiers	Word2vec	GloVe	Doc2vec
k-NN	**80**	**74**	81
NB	70	68	75
LR	79	72	82
SVM	78	**74**	82
SGD	76	50	76
DT	73	66	74
RF	78	72	76
MLP	**80**	73	**84**

Table 4 shows the F1-score of the CNN and RNN models. The CNN model attained 69% and outperformed RNN with significant difference. The validation set is part of the fitting of the model, so we also evaluated the performance of our model for different combinations of hyperparameter values. While validation accuracies of CNN are 83%, RNN has 67%. The dataset is also experimented on using the CNN-static and CNN-rand architectures developed by [29]. The results showed that a simple CNN architecture can achieve comparable results with CNN-static. CNN-rand gives better results than our CNN model. In such a case, all words are randomly initialized and then modified during training in CNN-rand, which does not use embedding weights.

Table 4. The F1-scores of CNN, RNN, CNN-static, and CNN-rand.

Models	F1
CNN	69
RNN	58
CNN-static	68
CNN-rand	74

In this study, MLP/Doc2vec outperformed simple word embeddings, BoW-tfidf (without feature selection), and also deep neural network architectures. It only underperformed BoW-tfidf using χ^2. Doc2vec can perform robustly when trained on larger corpora or can be further improved by employing the pretrained vectors.

Assessing the overall experiments, Table 5 shows that based on the experimental results, neural network architectures cannot show better performance than traditional ML methods with regard to author gender identification. Widely used algorithms such as k-NN, SGD, and MLP have proven to be efficient and successful for author gender identification.

Table 5. The best F1-scores in overall experiments.

Models	F1
SGD/BoW-tfidf-χ^2	91 (5K)/93 (10K)
k-NN/Word2vec	80
k-NN/Glove	74
MLP/Doc2vec	84
CNN	69
RNN	58

For feature representation, many studies want to use and compare both traditional representations such as BoW and embedding representations such as Word2vec or GloVe. Even though the drawback of the BoW approach is the curse of the dimensionality, the BoW approach has still the power to solve our gender problem. Feature selection also makes an important contribution to the system performance.

In recent years, unsupervised learned word embeddings models, particularly Word2vec and GloVe, have seen tremendous success and significantly outperformed distributional semantics models such as LSA, LDA, etc. However, these methods ignore sentiment information of texts and need a huge corpus. Normally, using pretrained word embeddings that were trained on other large text corpora increases the accuracy of the model.

We furthermore carried out analysis to investigate the impact of vector embedding dimensions. As

expected, we observed that system performance is definitely dependent on the size of the vector up to an optimum point. We checked the effect of the dimension by gradually changing the size from 100 to 500. Our results show that as the vector size increases, all the models show better performance for Word2vec, GloVe, and Doc2vec. The optimum value appears to be 300. Vector size has been discussed in many studies and the mostly preferred size is generally in the range of 100–500, depending on the problem and application domain.

Although the performance of deep neural networks depends on hyperparameters and training parameters such as dimension size, the size of the dataset is also important to achieve better results. When the size of the corpus is increased, naturally the results can be affected. In this study, the number of tokens in the dataset was 1.6M. Utilizing different very large datasets other than articles will be part of our further study. In addition, embeddings based on Word2vec, GloVe, and Doc2vec have been constructed from three Turkish corpora in this study. It could be important to determine whether using different Turkish embeddings (pretrained embeddings) instead of the current one would have a significant impact on the obtained performances. Some studies [23, 24, 35] employed combinations of CNN + long-short term memory (LSTM) or recurrent CNN to address the limitations of the RNN and CNN models for text classification. These studies provide insights to increase the accuracy of CNN and RNN models.

For the Turkish language, [27] proposed an n-gram model for identifying the gender of an author. The authors used C4.5, NB, SVM, and RF as classifiers with bi/trigram features and also CFS to obtain the feature subset. Although they showed that SVM outperformed all classifiers with 96.3% accuracy, cross-validation was not applied and the size of data was limited. The authors used 140 texts from females and 392 from males. To make a comparison, we ran another experiment with a similarly sized dataset as in [27]. We ran all the experiments with four more classifiers and achieved 96% with SGD in our study. Besides this result, the main contribution of our study is applying embeddings and complicated neural networks architectures to larger datasets.

6. Conclusion

In this study, we evaluated traditional ML and neural networks architectures to address the problem of author gender identification for Turkish. First, a list of ML classifiers was trained using several vector representations obtained by the variants of BoW and also distributed vector representations such as Word2vec, GloVe, and Doc2vec. We also trained and compared the results of a simple CNN and RNN.

A variety of experiments were conducted for the author gender identification problem in the Turkish language. The results indicated that the traditional ML algorithms outperformed CNN and RNN. While SGD/BoW-tfidf with χ^2 was found to be the best classifier with 91% F1-score for 5K feature size among ML methods, CNN achieved an F1-score performance of 69%. Doc2vec achieved a comparative performance with BoW-tfidf using MLP classifiers.

We also tested the system with regards to the size of dimensions, training architecture, and so forth. Hyperparameters and training parameters did not significantly improve the results. We did not observe any statistically important difference in dimension, batch size and training models such as, CBoW and SG.

In conclusion, this study is considered to be the first important attempt to use deep learning methods for the author gender identification problem in Turkish. Our proposed model is promising and gave successful results for this problem. As a future work, we will conduct some experiments using stylometric features and compare them with other deep neural network architectures such as LSTM and GRU with more datasets.

References

[1] Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. J Mach Learn Res 2003; 3: 1137-1155.

[2] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Machine Learning Research 2011; 12: 2493-2537.

[3] 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; 11–16 July 2010; Uppsala, Sweden. pp. 384-394.

[4] Mikolov T, Yih W, Zweig G. Linguistic regularities in continuous space word representations. In: Human Language Technologies Conference of the North American Chapter of the Association of Computational Linguistics; 9–14 June 2013; Atlanta, GA, USA. pp 746-751.

[5] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems; 5–10 December 2013; Lake Tahoe, NV, USA. pp. 3111-3119.

[6] Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. In: Empirical Methods in Natural Language Processing (EMNLP'14); 25–29 October 2014; Doha, Qatar. pp. 1532-1543.

[7] Le QV, Mikolov T. Distributed representations of sentences and documents. In: ICML'14 Proceedings of the 31st International Conference on International Conference on Machine Learning; 21–26 June 2014; Beijing, China. pp. 1188-1196.

[8] Wenpeng Y, Kann K, Yu M, Schütze H. Comparative study of CNN and RNN for natural language processing. CoRR, abs/1702.01923, 2017.

[9] Koppel M, Argamon S, Shimoni AR. Automatically categorizing written texts by author gender. Journal of Literary and Linguistic Computing 2002; 17: 401-412.

[10] 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: 169-175.

[11] Deitrick W, Miller Z, Valyou B, Dickinson B, Munson T, Hu W. Gender identification on Twitter using the modified balanced winnow. Journal of Communication and Network 2012; 4: 169-175.

[12] Burger JD, Henderson J, Kim G, Zarrella G. Discriminating gender on Twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11); 27–31 July 2011; Edinburgh, UK. pp. 1301-1309.

[13] Mukherjee A, Liu B. Improving gender classification of blog authors. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10); 9–11 October 2010; Massachusetts, USA. pp. 207-217.

[14] Argamon S, Koppel M, Pennebaker J, Schler J. Mining the blogosphere: age, gender and the varieties of self-expression. First Monday 2007; 12: 3.

[15] Schler J, Koppel M, Argamon S, Pennebaker J. Effects of age and gender on blogging, computational approaches to analyzing weblogs. In: 2006 AAAI Spring Symposium; 27–29 March 2006; Stanford, CA, USA. pp. 191-197.

[16] Cheng N, Chen X, Chandramouli R, Subbalakshmi KP. Gender identification from e-mails. In: IEEE Symposium on Computational Intelligence and Data Mining; 30 March–2 April 2009; Nashville, TN, USA. pp. 154-158.

[17] Cheng N, Chandramouli R, Subbalakshmi KP. Author gender identification from text. J Digital Investigation 2011; 8: 78-88.

[18] Alsmearat K, Al-Ayyoub M, Al-Shalabi R. An extensive study of the bag-of-words approach for gender identification of Arabic articles. In: Proceedings of IEEE/ACS International Conference on Computer Systems and Applications; 10–13 November 2014; Doha, Qatar. pp. 601-608.

[19] 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: IEEE/ACS 12th International Conference of Computer Systems and Applications; 17–20 November 2015; Marrakech, Morocco. pp. 1-6.

[20] Alsmearat K, Shehab M, Al-Ayyoub M, Al-Shalabi R, Kanaan G. Author gender identification from Arabic text. Journal of Information Security and Applications 2017; 35: 85-95.

[21] Bayot RK, Gonçalves T. Author profiling using SVMs and word embedding averages. In: Conference and Labs of the Evaluation Forum; 5–8 September 2016; Évora, Portugal. pp. 815-823.

[22] Markov I, Gómez-Adorno H, Posadas-Durán JP, Sidorov G, Gelbukh A. Author profiling with Doc2vec neural network-based document embeddings. In: Pichardo-Lagunas O, Miranda-Jiménez S, editors. Advances in Soft Computing. Cancún, Mexico: Springer, 2017. pp. 117-131.

[23] Sboev A, Litvinova T, Gudovskikh D, Rybka R, Moloshnikov I. Machine learning models of text categorization by author gender using topic-independent features. J Procedia Computer Science 2016; 101: 135-142.

[24] Sboev A, Litvinova T, Gudovskikh D, Rybka R. Deep learning network models to categorize texts according to author's gender and to identify text sentiment. In: International Conference on Computational Science and Computational Intelligence (CSCI); 15–17 December 2016; Las Vegas, NV, USA. pp. 1101-1106.

[25] Kucukyilmaz T, Cambazoglu BB, Aykanat C, Can F. Chat mining: predicting user and message attributes in computer-mediated communication. Journal of Information Processing & Management 2008; 44: 1448-1466.

[26] Can F, Patton JM. Change of word characteristics in 20th-century Turkish literature: a statistical analysis. Journal of Quantitative Linguistics 2010; 17: 167-190.

[27] Amasyalı MF, Diri B. Automatic Turkish text categorization in terms of author, genre and gender. In: Kop C, Fliedl G, Mayr HC, Métais E, editors. Natural Language Processing and Information Systems. Liège, Belgium: Springer, 2006. pp. 221-226.

[28] Leeuwenberg A, Vela M, Dehdari J, Genabith J. A minimally supervised approach for synonym extraction with word embeddings. Prague Bulletin of Mathematical Linguistics 2016; 105: 111-142.

[29] Yoon K. Convolutional neural networks for sentence classification. CoRR, abs/1408.5882, 2014.

[30] Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '13); 18–21 October 2013; Seattle, WA, USA. pp. 1631-1642.

[31] Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of SIGIR'15; 9–13 August 2015; Santiago, Chile. pp. 959-962.

[32] Wang X, Liu Y, Sun C, Wang B, Wang X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of ACL/IJCNLP; 26–31 July 2015; Beijing, China. pp. 1343-1353.

[33] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. P IEEE 1998; 86: 2278-2324.

[34] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing; 26–31 May 2013; Vancouver, Canada. pp. 6645-6649.

[35] Lai S, Xu L, Liu K, Zhao J. Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15); 25–30 January 2015; Austin, TX, USA. pp. 2267-2273.