Celal Bayar University Journal of Science

Deep Feature Generation for Author Identification

Şükrü Ozan[1]*, D. Emre Taşar[1], Umut Özdil[1]

[1] AdresGezgini Inc. Research and Development Center, Folkart Towers B Kule,
Office 3609, PK 35580, Bayraklı, İzmir, Türkiye.
*sukruozan@adresgezgini.com
*Orcid: 0000-0002-3227-348X

Received: 24 December 2020
Accepted: 3 May 2021
DOI: 10.18466/cbayarfbe.846016

Abstract

Identifying the authors of a given set of text is a well addressed and complicated task. It requires thorough knowledge of different authors' writing styles and discriminating them. As the main contribution of this paper, we propose to perform this task using machine learning and deep learning methods, state-of-the-art algorithms, and methods used in numerous complex Natural Language Processing (NLP) problems. We used a text corpus of daily newspaper columns written by thirty authors to perform our experiments. The experimental results proved that document embeddings trained via neural network architecture achieve cutting edge accuracy in learning writing styles and identifying authors of given writings even though the dataset has a considerably unbalanced distribution. We represent our experimental results and outsource our codes for interested readers and natural language processing (NLP) enthusiasts as a GitHub repository. They can reproduce and confirm the results and modify them according to their own needs.

Keywords: Natural Language Processing, Document Embeddings, Logistic Regression, Support Vector Machines, Author Identification.

1. Introduction

The rapid increase in the number of digital texts has triggered academic research to identify and verify the authors from a given set of a text corpus. By applying computational learning approaches, authors' writing styles and their thematic interests can be captured with an accuracy comparable with human-level performance (HLP). For a predefined set of given authors, determining the most probable author for the given text is author identification [1] that can be considered as a multi-class text categorization problem from a machine learning (ML) and deep learning (DL) perspective [2]. The author identification is generally performed in a closed domain. However, the numbers of texts from selected authors do not always have to be balanced [3].
ML/DL based author identification can effectively be used in disputed authorship cases [4] and literary analysis studies [5]. Like in any other DL study, the heterogeneous distribution of sample data is a problem for author identification. This problem is addressed thoroughly in [6]. Despite the significant amount of work devoted to author identification of a text, researchers still struggle to deal with cross-domain texts and imbalanced datasets.

One of the fundamental steps in author identifications is stylometry, which refers to discovering the most specific features representing an author's writing characteristics [7]. Earlier approaches focused on generating metrics to describe function word or part-of-speech frequencies to assess the vocabulary's diversity. A detailed review of these approaches is represented in [8] to highlight the importance of extracting features.
In this study, the author identification framework is proposed to generate a text's specific features by combining ML and DL methods commonly used in natural language processing (NLP) literature. The author identification procedure is performed by discovering features, thematic interests, unique characteristics of an author, and writing styles. Most accurate results are achieved with Doc2Vec D-BOW [9,10] model with a C-SVC [11] classifier.

The rest of the paper is organized as follows: Section 2 describes the materials and methods applied for this study. Experiments and results are presented in Section 3. Section 4, as the final section, conclusions and future research suggestions are given.

2. Materials and Methods

2.1 Dataset

A famous Turkish news portal, subsuming more than 50 authors, has been scrutinized to build a corpus. The authors are of different ages, genders. Moreover, their writing themes are from different genres, such as politics, sports, health, and literature. We randomly picked 30 of the authors. Figure 1 shows the number of articles per author, which represents an unbalanced distribution. We intentionally kept the author names hidden and used numbers representing Author IDs. After selecting the authors, we scraped the website and gathered all of their articles labeled with author IDs.

2.2 Data Preprocessing

Data pre-processing is an essential step in NLP methods since it is vital to use clean text data that is free from characters and symbols that may introduce bias and errors during the learning process. The data is scraped directly from the news portal website; hence, it has punctuation symbols and many HTML related tags. There are various text pre-processing methods, including but not limited to converting capital letters to lowercase letters (case folding), clearing symbols, and punctuation marks [12]. We used a 70-30 split scheme for training and validation data sets after the data pre-processing operations.

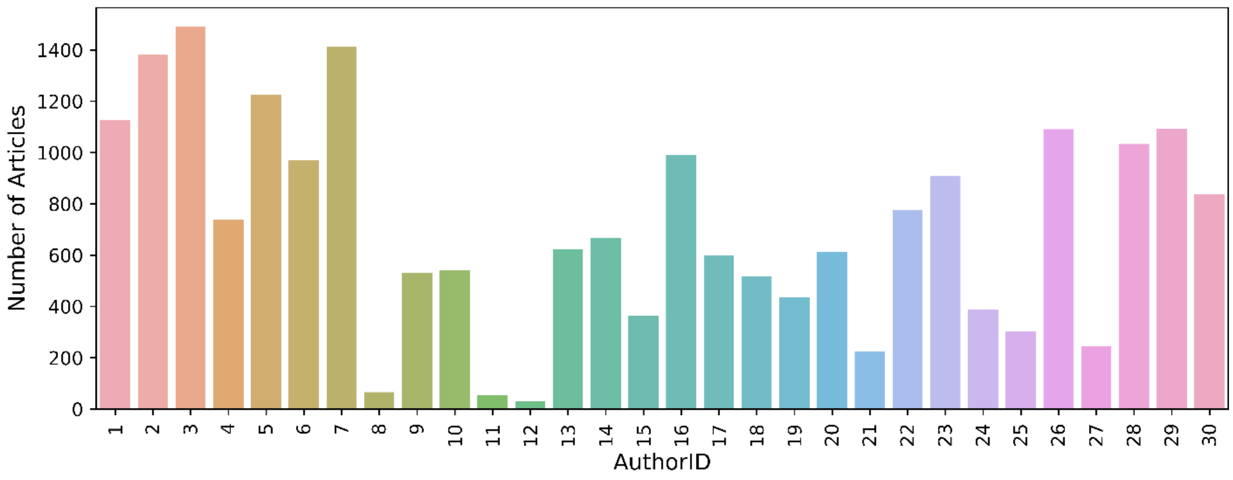


Figure 1. The number of articles per author can be seen in this figure. Authors' names are intentionally hidden and represented as ID numbers.

2.3 Doc2Vec Embeddings

Usage of using Doc2Vec for text classification tasks has gained well-deserved popularity in NLP literature. We used the Gensim [13] implementation of the Doc2Vec method [14]. The method can generate the same embedding sizes for input text with different sizes. The method relies on two main learning models: Distributed Memory (DM) and Distributed Bag of Words (D-BOW). In our experiments, we used both models and compared their performances in the results section.

2.3.1 Distributed Memory (DM)

DM model, depicted in Figure 2a., has a similarity with the Skip-Gram method of Word2Vec [14]. The CBOW model can predict a center word based on the context words in a small neighborhood. DM uses a similar approach to randomly sample some context words from an article and predict a word using both the context words and the article ID.

2.3.2 Distributed Bag of Words (D-BOW)

D-BOW model, depicted in Figure 2b., uses the whole article as input and tries to predict consecutive Word chunks from the article having a meaningful context. One advantage of the D-BOW model is that it requires less memory during training time. Activation function weights are enough to be stored for D-BOW to operate.

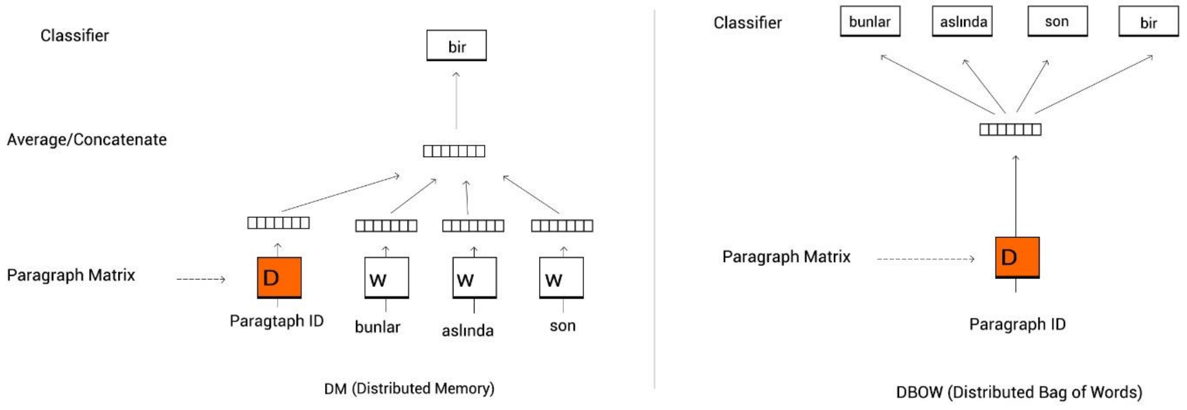


Figure 2. a) Distributed Memory model, where classifier predicts a word as an output parameter and words concatenate as word vectors and paragraph vector. b) Distributed Bag of Words model, where paragraph vector is trained to predict the words inside a small window.

2.4 Classifiers

With either model, DM, and D-BOW, the Doc2Vec method generates a fixed-size embedding vector for each article. These vectors can capture the meaning, syntax, writing style, and other linguistic features of a given text in hyperdimensional space.

Together with the author ID information, the embeddings can be used to train a classifier. In this study, we preferred two different classifiers. Conventionally logistic regression classifier (LRC) is a common choice for the classification of multidimensional data [15]. LRC is proven to be an adequate method, especially for binary classification problems where the number of classes is limited (i,.e. N = 2). On the other hand, our problem is a multi-class classification problem with 30 classes (i.e., N > 2). Multinomial logistic regression classifier MLRC [16] generalizes the binary classification idea of standard LRC to multi-class classification. Once the coefficients are determined after running the MLRC, the probability of predicting an author's class can be done using Equation 2.1, where k is the class number, x is the embedding vector, and β_k is the coefficient vector of class k. Hence, MLRC models the probability of a given data belonging to a class using log-likelihood estimation as given in Equation 2.1.

$$L = \log \mathrm{P}(y = k \mid x_i) \quad \text{where}$$
$$\mathrm{P}(y = k \mid x_i) = \frac{\exp^{\beta^k x}}{1 + \sum_{k=1}^{K-1} \exp^{\beta^k x_k}} \qquad \textbf{(2.1)}$$

We used MLRC as our first classifier, and it achieved good accuracies on both training and validation datasets. However, we repeated our experiments also by using support vector machines (SVM) based classifier (C-SVC) [11]. This classifier creates hyperplanes in a multidimensional space that can be efficient for classification and regression tasks.
It finds the best hyperplane that demonstrates the largest segregation between the two classes. The error of this classifier is correlated with the size of the margin [11]. C-SVC uses kernels with different mathematical models. For this study, we used the linear kernel, in Equation 2.2, where K is the kernel function, D is the decision function, y_i, b, and α_i are indicator vector, equation constant, and dual parameter, respectively. C-SVC classifier achieved even better results than MLRC,

$$K(x_i, x_j) = x_i^T x_j \qquad \textbf{(2.2)}$$
$$\mathrm{D} = \operatorname{sgn}.(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b)$$

2.5 F1 Score as an Accuracy Metric

To better identify the algorithms' classification accuracy, we calculated the F1 Score for each class. F1 Score, which calculates harmonic mean between recall

$$\Pr = \frac{TP}{TP + FP} \qquad \textbf{(2.3)}$$
$$\mathrm{Re} = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{\Pr * \mathrm{Re}}{\Pr + \mathrm{Re}} \qquad \textbf{(2.4)}$$

$$F_{avg} = \frac{2 * P_M * R_M}{P_M + R_M} \qquad \textbf{(2.5)}$$

and precision values, can be calculated using Equation 2.4; in this equation, TP, FP, and FN represent true positive, false positive, and false negative values, respectively. Definition of Precision (Pr), Recall (Re) can be calculated using Equation 2.3. The F1-Average score can be calculated using Equation 2.5 for each class. In this equation, macro average values are represented with M. Hence, P_M and R_M represent macro averaged precision and recall values.

3 Results and Discussion

From the NLP perspective, the classification task becomes challenging where the number of classes exceeds 20. Moreover, the data's unbalanced nature makes it even harder for an ML/DL-based algorithm to achieve a good generalization for the whole dataset. In our study, we have both situations.

We started our experiments with hyperparameter tuning of the Doc2Vec model. The parameters, which affect the result most are the embedding size and the number of epochs. By keeping the remaining model parameters constant, we tried different embedding sizes to detect the optimal size. The constant parameters and their default values can be seen in Table 1. We observed the classification accuracy in Table 2 after one epoch of training by trying different embedding sizes.

Table 1. Constant Parameters for Doc2Vec Model

Model Parameters*	**Value**	**Model Parameter**	**Value**
window	10	alpha	0.0061
negative	5	Min_alpha	0.0001
Min_count	1		

* Other model parameters set to default

We used MLRC to calculate the training and validation accuracies. Since the number of classes is considerably high, we achieved the optimum result with a 500-dimensional embedding size. We did not get a significant performance increase after 500. After fixing the embedding size to 500, we tried different epochs and different classifiers. The results of these experiments are given in Table 3. We further observed that both DM and D-BOW methods start to overfit after a few epochs. Hence, we limited the number of epochs to 10 for each experiment.

As it can be interpreted by looking at Table 3., we achieved the best accuracies for the validation data set for 500-dimensional vector size at the 4th epoch of D-BOW training using the C-SVC classifier. It is also possible to grasp the success of this result by visually examining the embedding vectors. Since the embeddings are in hyperspace, we used the UMAP projection method [17] to visualize them in 2D. Figure 3 shows how the embeddings belonging to seven of the authors are successfully clustered together.

The overall clustering performance of the algorithm can also be shown with a confusion matrix. In Figure 4, the confusion matrix of the classification for 30 authors can be seen. The confusion matrix can also be used to calculate the F1 Score for each of the classes. In Table 4., we listed the calculated F1 Score for each class. For authors 8 and 12, the worst classification results are achieved. These authors have significantly less number of articles compared with the remaining authors. However, we can also see that for another author (author 11) with few articles; high accuracy is achieved. When we closely examine these three authors, we saw that author 11 has a particular subject genre compared with the other authors.

On the other hand, authors 8 and 12 mostly write interview articles where they mostly quote the person being interviewed, making it harder for the algorithm to learn and generalize for these specific authors. If we disregard authors 8 and 12 and only rely on the remaining 28 authors, the training and validation accuracies hit 1.00 and 0.97, respectively. F1 Scores and their weighted averages are found to be 0.98 and 0.97 using Equation 2.3 - 2.5.

Table 2. Train and Validation Accuracies for Vector Sizes

Model	**D-BOW + MLRC**		**DM + MLRC**	
Accuracy	**Train**	**Validation**	**Train**	**Validation**
5	0.7716	0.7453	0.6737	0.6704
25	0.9777	0.9496	0.8602	0.8427
50	0.9904	0.9503	0.9073	0.8823
100	0.9951	0.9516	0.9392	0.9002
300	0.9953	0.9596	0.9646	0.9209
500	**0.9936**	**0.9581**	0.9626	0.9225
1000	**0.9891**	**0.9589**	0.9555	0.9243

Table 3. This table shows the change in Training and Validation Accuracies for Different Doc2Vec Method and Classifier Combinations for different epoch numbers. All the model combinations tend to overfit the data after a few epochs of training.

Model	**D-BOW MLRC**		**DM MLRC**		**D-BOW C-SVC**		**DM C-SVC**	
Accuracy	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**	**Train**	**Validation**
Epoch 1	0.9955	0.9601	0.9689	0.9272	0.9685	0.948	0.7968	0.7811
Epoch 2	0.9996	0.9543	0.9966	0.9425	0.9937	0.9654	0.9615	0.9272
Epoch 3	0.9999	0.9486	0.9998	0.9369	0.9967	0.9667	0.9836	0.9433
Epoch 4	**1**	0.9519	**1**	0.9373	0.9975	**0.9682**	0.9894	0.9501
Epoch 5	**1**	0.948	**1**	0.938	0.9968	0.9663	0.991	0.9484
Epoch 6	**1**	0.9501	**1**	0.9386	0.9956	0.9669	0.9879	0.9507
Epoch 7	**1**	0.9402	0.9977	0.8573	0.9927	0.9593	0.9555	0.8814
Epoch 8	0.9995	0.9325	0.9651	0.7737	0.9842	0.9427	0.8669	0.7524
Epoch 9	0.6226	0.4788	0.4	0.2454	0.5175	0.4487	0.2689	0.2314
Epoch 10	0.5802	0.4945	0.7323	0.516	0.5367	0.4598	0.5871	0.4838

Table 4. D-BOW + C-SVC Pr, Re, F1 Score Results

Authors	**Precision**	**Recall**	**F1-Score**	**#Articles**	**Authors**	**Precision**	**Recall**	**F1-Score**	**#Articles**
1	0.97	0.99	0.98	514	16	0.91	0.92	0.91	390
2	0.99	0.98	0.98	625	17	**1.00**	0.99	0.99	247
3	0.98	1.00	0.99	653	18	0.97	0.98	0.97	237
4	0.97	0.96	0.96	281	19	0.97	0.97	0.97	183
5	0.98	**1.00**	0.99	562	20	0.95	0.95	0.95	293
6	0.97	0.97	0.97	397	21	0.98	0.98	0.98	116
7	0.97	0.97	0.97	645	22	0.99	0.98	0.98	329
8	0.70	0.91	0.79	33	23	0.99	0.99	0.99	361
9	0.98	0.95	0.97	264	24	0.99	0.97	0.98	159
10	**1.00**	0.98	0.99	243	25	**1.00**	**1.00**	**1.00**	141
11	0.96	0.82	0.89	33	26	0.98	0.98	0.98	462
12	0.68	0.71	0.70	21	27	0.95	0.94	0.94	112
13	0.86	0.92	0.88	226	28	0.95	0.99	0.97	389
14	0.98	0.91	0.94	274	29	0.95	0.95	0.95	415
15	0.98	0.86	0.91	162	30	0.98	0.97	0.97	354
			Accuracy	**0.97**	**0.97**	**0.97**	**9121**		
			W. Avg	**0.97**	**0.97**	**0.97**	**9121**		

.

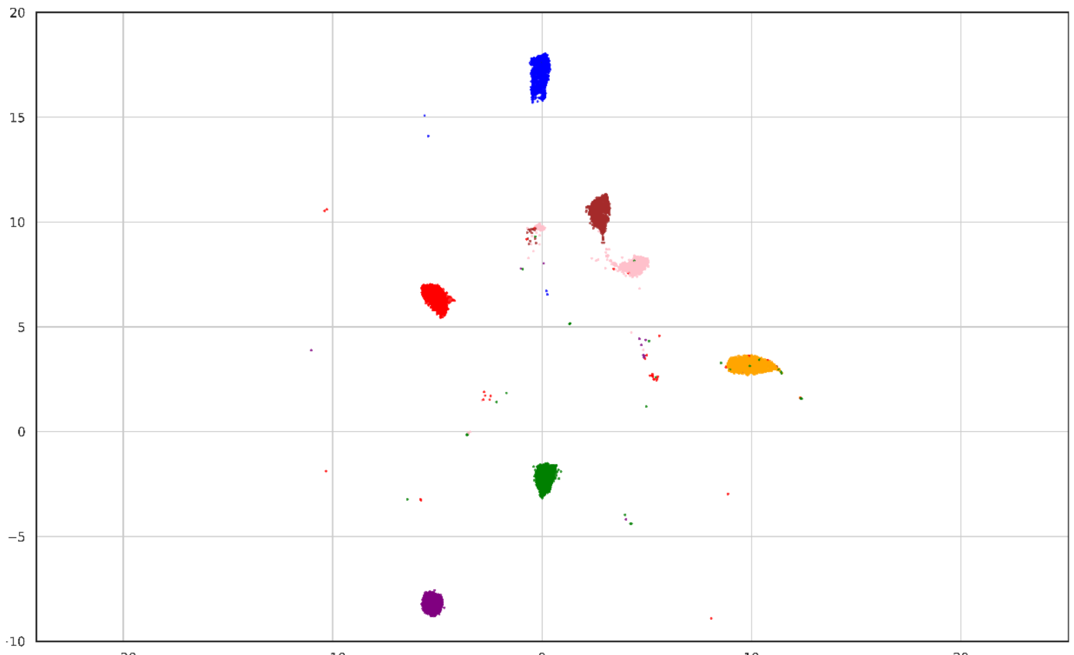
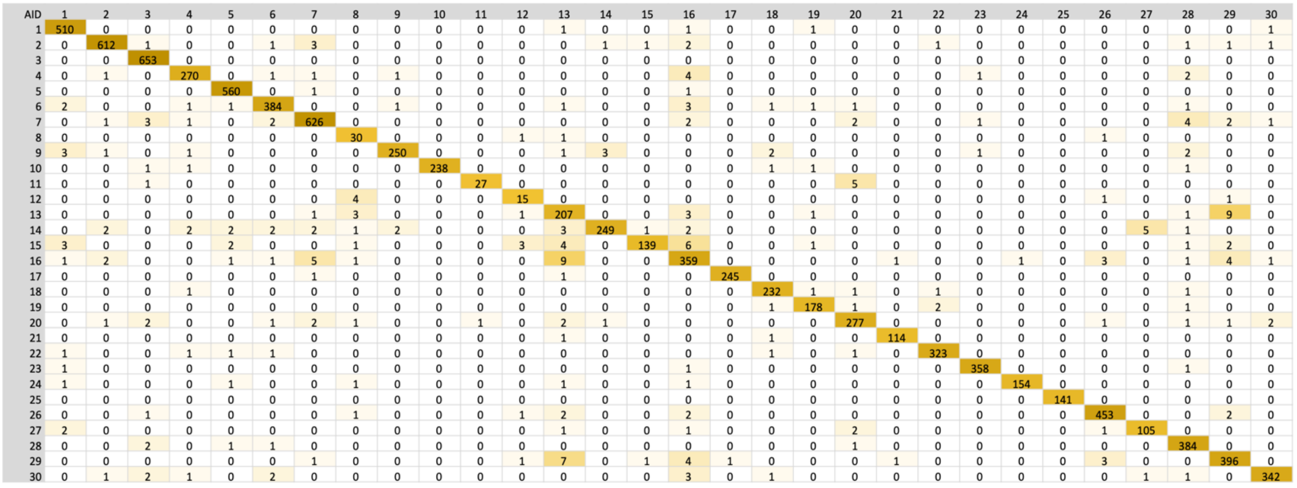


Figure 3. 2D UMAP projection of the training data's embedding vectors created with the best configuration in Table 3. For display purposes, we zoomed the area where the clusters of seven authors are visible together.



AID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	510	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
2	0	612	1	0	0	1	3	0	0	0	0	0	0	1	1	2	0	0	0	0	0	1	0	0	0	0	0	1	1	1
3	0	0	653	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	270	0	1	1	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0	0	2	0	0
5	0	0	0	0	560	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	0	0	1	1	384	0	0	1	0	0	0	1	0	0	3	0	1	1	1	0	0	0	0	0	0	0	1	0	0
7	0	1	3	1	0	2	626	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	1	0	0	0	0	4	2	1
8	0	0	0	0	0	0	0	30	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
9	3	1	0	1	0	0	0	0	250	0	0	0	1	3	0	0	0	2	0	0	0	0	1	0	0	0	0	2	0	0
10	0	0	1	1	0	0	0	0	0	238	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0
11	0	0	1	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	4	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
13	0	0	0	0	0	0	1	3	0	0	0	1	207	0	0	3	0	0	1	0	0	0	0	0	0	0	0	1	9	0
14	0	2	0	2	2	2	2	1	2	0	0	0	3	249	1	2	0	0	0	0	0	0	0	0	0	0	5	1	0	0
15	3	0	0	0	2	0	0	1	0	0	0	3	4	0	139	6	0	0	1	0	0	0	0	0	0	0	0	1	2	0
16	1	2	0	0	1	1	5	1	0	0	0	0	9	0	0	359	0	0	0	0	1	0	0	1	0	3	0	1	4	1
17	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	245	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	232	1	1	0	1	0	0	0	0	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	178	1	0	2	0	0	0	0	0	1	0	0
20	0	1	2	0	0	1	2	1	0	0	1	0	2	1	0	0	0	0	0	277	0	0	0	0	0	1	0	1	1	2
21	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	114	0	0	0	0	0	0	0	0	0
22	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	323	0	0	0	0	0	0	0	0
23	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	358	0	0	0	0	1	0	0
24	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	154	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	141	0	0	0	0	0
26	0	0	1	0	0	0	0	1	0	0	0	1	2	0	0	2	0	0	0	0	0	0	0	0	0	453	0	0	2	0
27	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	0	1	105	0	0	0
28	0	0	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	384	0	0
29	0	0	0	0	0	0	1	0	0	0	0	1	7	0	1	4	1	0	0	0	1	0	0	0	0	3	0	0	396	0
30	0	1	2	1	0	2	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	1	1	0	342

Figure 4. Confusion Matrix of classification performed using the best configuration in Table 3 for 30 author classes' validation data

.

4 Conclusion

For this study, we scraped 30.403 articles of randomly selected 30 different authors of a popular Turkish news portal. After the pre-processing of raw page source data, we obtained cleaned text for each article. After tagging each article with corresponding author IDs, we obtained our dataset. We used a 70-30 split scheme for training and validation data set. The data set is considerably unbalanced, i.e., the variance in the number of articles is high.

The proposed algorithm mainly relies on the Doc2Vec method, which uses two different learning models: distributed memory and distributed bag of words. Regardless of the preferred model, this method generates a fixed size embedding vector for given texts of different sizes. We calculated the models' training and validation accuracy at each training epoch after applying two different classifiers, MLRC and C-SVC. Each of the model and classifier combinations gave good accuracies comparable with HLP. Hence, it is possible to obtain deep features for author classification with the proposed solution methods. The method can detect possible plagiarisms in closed domains, such as a corpus of homework reports submitted by students.

As future work, it is possible to train different convolutional neural networks similar to MGNC-CNN architecture in [18] using these deep feature embeddings, which may yield good classification accuracies. The deep features can further be used to train recurrent generative adversarial networks [19], generating artificial texts that mimick the corresponding authors' writing styles.

We are outsourcing our code [20] for NLP researchers and enthusiasts to reproduce the reported results and use them in their research.

Acknowledgment

This work is a part of the project supported by the Scientific and Technological Research Council of Turkey (TUBITAK) TEYDEB-1501 program under Project no 3190585, and named "General Purpose Chatbot Application That Can Produce Meaningful Dialog via Machine Learning Algorithms".

Author's Contributions

Şükrü Ozan: Built the algorithms and metric criteria, prepared the web scraping scripts, drafted and wrote the manuscript.
D. Emre Taşar: Supervised the experiments, helped in result interpretation and literature review, assisted in manuscript preparation.
Umut Özdil: Prepared the text pre-processing scripts, assisted in data visualization.

Ethics

The Ethics Committee has ruled that approval was not required for the study.

References

1. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: 2000. Automatic text categorization in terms of genre and author. *Comput. Linguist. 26(4),* 471–495

2. Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys,* 34(1): 1-47.

3. Zheng, Rong, et al. 2006. "A framework for authorship identification of online messages: Writing-style features and classification techniques." *Journal of the American society for information science and technology* 57.3 : 378-393.

4. Burrows, J.F. 1987. Word Patterns and Story Shapes: The Statistical Analysis of Narrative Style. *Literary and Linguistic Computing* 2: 61-70.

5. Diederich, J., J. Kindermann, E. Leopold, and G. Paass. 2003.. Authorship Attribution with Support Vector Machines. *Applied Intelligence 19(1/2): 109-123*

6. Luyckx, K., Daelemans 2011, W.: The effect of author set size and data size in authorship attribution. *Literary Linguist. Comput.* 26(1), 35–55

7. Abbasi, Ahmed, and Hsinchun Chen. 2008. "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace." *ACM Transactions on Information Systems* (TOIS) 26.2 : 1-29.

8. Holmes, D. 1998. The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguistic Computing,* 13(3): 111-117.

9. Mikolov, Tomas, et al.2013. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*.

10. Mikolov, Tomas, et al. 2013. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 26: 3111-3119.

11. Cortes, Corinna, and Vladimir Vapnik.1995. "Support-vector networks." *Machine learning* 20.3: 273-297.

12. Li, J., Huang, G., Fan, C., Sun, Z., & Zhu, H. (2019). Key word extraction for short text via word2vec, doc2vec, and textrank. *Turkish Journal of Electrical Engineering & Computer Sciences*, 27(3), 1794-1805.

13. Rehurek, R., Sojka, P. 2010. Software framework for topic modelling with large corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, pp.* 45–50.

14. Kim, Donghwa, et al. 2019 "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec." *Information Sciences* 477 : 15-29.

15. Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M. Ingersoll. 2002. "An introduction to logistic regression analysis and reporting." *The journal of educational research* 96.1: 3-14.

16. Kwak, Chanyeong, and Alan Clayton-Matthews.2002 "Multinomial logistic regression." *Nursing research* 51.6: 404-410.

17. Becht, Etienne, et al. 2019. "Dimensionality reduction for visualizing single-cell data using UMAP." *Nature biotechnology* 37.1 : 38-44

18. Zhang, Ye, Stephen Roller, and Byron Wallace. 2016. "MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification." *arXiv preprint arXiv:*1603.00968 .

19. Radford, Alec, Luke Metz, and Soumith Chintala. 2015 "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:*1511.06434

20. "Deep Feature Generation for Author Identification " https://github.com/adresgezgini/DFG4AI/