**Column-Author Matching in Turkish Texts Using**

**SVM and MLP Algorithms**

***İbrahim DOĞAN1,[[1]](#footnote-1)\* [C:\Users\Abdullah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ORCID-iD_icon-16x16.gif](https://orcid.org/0000-0003-0412-5559), Sedat KORKMAZ 2[C:\Users\Abdullah\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ORCID-iD_icon-16x16.gif](https://orcid.org/0000-0002-7690-5979)***

*1 Institute of Graduate Studies, Faculty of Engineering and Natural Sciences, Computer Engineering, Konya Technical University, Konya/Turkey*

*2* *Institute of Graduate Studies, Faculty of Engineering and Natural Sciences, Computer Engineering, Konya Technical University, Konya/Turkey*

|  |
| --- |
| **Abstract**  In this study, a column-author matching was performed using columns taken from the website of a newspaper. A data set containing more than 3 thousand data was created by combining the articles written by people who are newspaper writers in different fields, some other information about the articles and the names of the authors. Data pre-processing work was carried out on the created data set and data set optimisation was achieved. By running Zemberek library functions on the optimised data set, morphological analysis was performed, stop words were removed from the text and new attributes were extracted and included in the data set. The data was normalized using the Min-Max Normalization technique and digitised with TF-IDF, Word Bag and Word-Sentence Distribution Vectors. The 67% of the dataset was used to train the machine learning classification algorithms, while the remaining 33% was used to measure the author-artwork matching success of the algorithm. Support Vector Machines (SVM) and Multilayer Perceptrons (MLP) were used in the study and the highest success rate was obtained with the SVM algorithm with 83%. |
| Keywords: Machine Learning, Natural Language Processing, Text Classification, Zemberek Library. |

1. **Introduction**

Language is an advanced communication tool that enables people to express their feelings and thoughts using words or signs. Languages are categorized under two main headings: natural languages and artificial languages.

Natural language is a decodable multi-level system with a certain rule structure that people use in their daily lives to share their ideas, wishes and thoughts with each other. English, German, Spanish, French and Chinese are examples of natural languages that are widely used in the world.

Artificial languages, on the contrary to natural languages, are languages whose source is known, created and developed by some people or communities for a purpose. It is possible to analyze artificial languages in two different categories. These are programming languages used in computer science (Java, C#, C++, Python, HTML, etc.) and languages used by some communities as spoken languages (Esperanto, Bâleybelen, Ido, Elvish, Klingon, etc.).

When the history of humanity is analyzed, the leaps provided by some developments draw attention. Language is one of these developments. Thanks to language, changes and progress have been made in many fields such as faith, culture, art and positive sciences. Today, many applications are being developed in the field of Natural Language Processing (NLP) due to the intensification of studies in the field of machine learning, linguistics and artificial intelligence, the rapid development of technology and the increase in human-computer interaction.

Many sciences and disciplines are utilized in the development of NLP applications. NLP is a bridge that works interactively with the fields of artificial intelligence, linguistics, cognitive psychology and data science and establishes the relationship between machine language and spoken language by taking text or audio data as input and providing the desired outputs within certain rules. NLP is a combination of Natural Language Understanding (NLU) and Natural Language Generation (NLG) processes [1]. Although we are not aware of it, NLP is frequently encountered in our daily lives. For example; voice response systems, text summarisation applications, spell checking and word suggestion systems are just some of the areas of use. Considering the studies carried out, it is better understood how intertwined NLP is with our lives and how it is a factor that facilitates our lives.

The working steps of NLP are generally analyzed under four main headings. These are respectively; Vocabulary Science, Syntactic Analysis, Semantic Analysis and Discourse Analysis. In the morphological analysis stage, the smallest structural units of words, roots and affixes, are examined separately to determine the type and form. In the syntactic analysis stage, the relationship between the words in the sentence is interpreted. Semantic analysis is the method used to determine which feelings and thoughts are intended to be expressed in the sentences in the texts. Discourse analysis is the method aiming to analyse the main idea of the text correctly.

The rapid development of technology and the widespread use of the internet has led to the provision of many services in this field, easy access to data and the number of data has increased exponentially day by day. This situation has brought with it the difficulty of accessing the correct and desired data. For this reason, especially the extraction of meaningful data from meaningless data and the classification of such data according to certain qualities have gained importance.

In this study, the articles written by newspaper writers in different fields were analyzed by using Numerical Style Analysis (Stylometry), NLP, a sub-branch of artificial intelligence, and machine learning methods, and it was aimed to accurately match the authors of the articles in question.

1. **Materials and Methods** 
   1. **Software Packages**

The coding of the project was done using Anaconda distribution version 1.9.0, Python version 3.9.12 and Jupyter Notebook version 6.4.11 development environment. The Java-based Zemberek library version 0.17.1 was downloaded as a Jar file, the Zipfile library was used to read the modules in the Jar file, and version 1.3 of the Jpype library was used to integrate it into Python. Version 1.4.1 of the Pandas library was used for data manipulation, version 1.20.3 of the Numpy library was used for high-level mathematical operations, and version 1.0.2 of the Scikit-learn library was used for machine learning models and some preprocessing modules.

* 1. **Data Set**

The data set was obtained from the digital platform of a newspaper using Web Scraping method. There are 208 columnists writing in different fields (agenda, sports, market, etc.) in the dataset. The dataset includes a total of 3417 columns, with each columnist having at least 1 and at most 40 columns. The data are kept in table format and consist of the columns of the date the column was written, the name of the author, the headline of the column, the link to the newspaper website from which the column was taken and finally the text of the column.

* 1. **Data Preprocessing**

Data preprocessing is defined as a set of methods that should be applied before starting to work with the data set in order to ensure that machine learning algorithms can benefit from the data set at the highest level and obtain the most accurate results [2, 3]. The data preprocessing techniques used in this study are listed below.

* + 1. **Cleaning Stop Words from the Data Set**

Insignificant words that are used for grammatical rules or to ensure sentence integrity and do not carry any information about the text they are found in are labelled as stop words. These stop words are removed from the data set in the data preprocessing stage in order to increase the classification success and reduce the processing time and data size.

* + 1. **Min-Max Normalization**

Normalization is used to reduce data size, to reduce data clutter by formatting within a scale without proportionally distorting the difference between very distant numerical values, to make the data set more meaningful and interpretable and to provide better results of machine learning applications [4].

In this study, the data set is scaled between 0 and 1 using the Min-Max Normalization technique. The minimum is the smallest value that the data can take and the maximum is the largest value that the data can take [5]. The Min-Max Normalization technique was performed using Equation 1. [6, 7].

(1)

* 1. **Zemberek**

Zemberek was developed by Ahmet Afşın AKIN using Java programming language and it is a library that enables the application of NLP methods on Turkish texts [8]. With the help of the library, operations such as spell checking, sentence separation, correction of misspelled words, detection of written language, finding word roots and affixes can be easily performed.

* + 1. **Morphological Analysis**

Morphological analysis is defined as the separation of roots and affixes of a word in order to determine its morphological structure. Since Turkish is a suffixal language, the suffixes to the word can cause changes in the structure of the word and change its meaning. For this reason, it becomes difficult to find the root and affixes of the word in suffixal languages such as Turkish [9].

In the Zemberek library, there is a special tree model designed to find the roots of Turkish words. The roots are placed and labeled on the tree according to their letter-based content. For example, the noun root "balo" is added to the tree and labelled with the letters B-A-L-O respectively. One of the most important advantages of the model is that it can be used by using or adding existing nodes without having to create separate nodes for each root. Thus, unnecessary nodes are not used and memory is saved [10].

* + 1. **Sentence Extraction/Splitting**

Zemberek library can split the text into its sentences as well as its roots and suffixes. TurkishSentenceExtractor class is used for this. The class takes a text as input and returns separated sentences of the text as output [11]. At this stage, rule-based simple combinations and the Binary Averaged Perceptron model are used to determine the start and end points of the sentence.

* 1. **Term Frequency-Inverse Document Frequency (TF-IDF)**

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical method used to determine the importance of a term in a text within a dataset. TF refers to the frequency of the relevant term in the processed text and is calculated using Equation 2. When calculating the TF value, all words in the text are considered equally important [12].

(2)

Document Frequency (DF) is the frequency of the related word in other texts and IDF (Inverse Document Frequency) is the logarithm of the DF value. IDF is a factor that decreases the weight of terms that occur frequently in the whole document and therefore have a higher frequency than other words in the text, and increases the weight of words that rarely occur in the document with low frequency and are more important for the document. As seen in Equation 3, the IDF value is calculated by taking the logarithm of the result obtained by dividing the total number of documents by the number of documents in which the relevant word occurs [13].

(3)

TF-IDF is a frequency obtained by multiplying these two values and varies between 0 and 1. A higher TF-IDF value indicates that the word is more important for the text and represents the text more.

* 1. **Classification Algorithms**

This study employed two distinct classification algorithms, which are described below.

* + 1. **Support Vector Machine (SVM)**

SVM is a supervised machine learning algorithm used to solve classification and regression problems. The method utilizes vectors to separate data clustered around certain attributes. These vectors separating the classes are characterised as hyperplanes. In order to increase the success of the classification, SVM prefers the vector that is the farthest distance from the data of all classes. This preferred vector is called the optimum hyperplane. The points or data closest to the optimum hyperplane are called support vectors. The gap between the supports is called Margin.

* + 1. **Multi-Layer Perceptron (MLP)**

The MLP algorithm consists of 3 main parts. These are respectively: Input Layer, Hidden Layer and Output Layer.

The number of neurons in the input layer is equal to the length of the vector received as input. The number of neurons in the output layer is designed to be equal to the length of the vector to be obtained as a result of the model. The intermediate layers, consisting of at least one layer between the input and output layers, are called hidden layers. There should be at least one neuron in the hidden layer, but one neuron may not be enough to achieve the desired success rate. For this reason, various experiments are performed to determine the optimal number of neurons in the hidden layer of the model and the design that gives the most successful result is applied to the model. In other words, the number of hidden layers and the number of neurons in the hidden layer are determined by trial and error [14].

* 1. **RandomizedSearchCV**

In order to ensure that classification algorithms can work with maximum performance, parameter adjustment is needed. The number of parameters may vary according to the algorithms. However, depending on the technical characteristics of the data set used in the study, many parameters may need to be adjusted. In such cases, it is very unlikely that the optimal parameter set can be found manually due to the large number of possible combinations. Therefore, methods have been developed to find the optimal parameters for algorithms. In this study, the RandomizedSearchCV method is used, which is often preferred for hyperparameter analysis. The RandomizedSearchCV method tests a certain number of randomly selected parameter combinations on the data set. In the RandomisedSearchCV method, the number of iterations is determined by the user in order to find the most successful hyperparameter set.

1. **Results and Discussion**

This study consists of 3 basic process steps. These are; data collection and creation of the dataset, feature extraction from the pre-processed data and finally classification of the dataset using machine learning algorithms.

The dataset was obtained from a newspaper's website using the Requests and BeautifulSoup libraries. While news articles written in different fields were collected for data diversity, the last 40 articles of the authors (if the columnist has less than 40 published articles, all of their existing works were taken) were added to the dataset to ensure the timeliness of the news.

Since some of the data in the obtained data set showed an unbalanced distribution, 96 authors with less than 15 columns and 330 columns of these authors were considered as outlier data and removed from the data set. As a result of this process, a total of 112 authors and 3087 columns remained in the dataset.

Using Zemberek, an NLP library, the texts in the dataset were separated into words and sentences, stop words were cleaned and word roots were extracted. In addition, numerical attributes such as word richness, root richness, punctuation marks used and number of stop words were extracted and added to the dataset. In addition, the numerical attributes were scaled using the Min-Max Normalization method in order to maximize the classification success.

In text-based classification studies, the most information about the scope of the text is usually contained in word roots. For this reason, word roots have an important place as a distinguishing feature in text classification applications. However, since most of the classification algorithms cannot work on categorical data, such data must be transformed. For this reason, the word roots in the dataset were digitised using TF-IDF and Word Bag methods.

The data set, which was made suitable for classification, was divided into two parts as training (67%) and test (33%) and SVM and MLP algorithms were trained on the training data set. The trained classification algorithms were run on the test data with the default parameters and the success rates were calculated. As a result of this process, the MLP algorithm achieved a success rate of 79.2%, while the SVM algorithm achieved a success rate of 65.5%.

In order to further increase the success rates, hyperparameter analysis was performed with the RandomisedSearchCV method and reclassification was performed using the optimum parameters obtained. As a result of the classification, a success rate of 83% was obtained with the SVM algorithm, while a success rate of 82.4% was obtained with the MLP algorithm. The classification success rates of these algorithms before and after hyperparameter analysis are given in Table 1. In the hyperparameter analysis phase, the number of iterations for the RandomizedSearchCV method was set to 20.

**Table 1.** Success rates obtained before and after hyperparameter analysis.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Default parameters | Success Rates | Parameter Pool | The Best Parameters | Success Rates After Parameter Analysis | |
| SVM | Kernel=rbf  C= 1.0 | %65.5 | Kernel=[rbf, linear, sigmoid, poly]  C= Between(1-1000)  Gamma= Between (0.0001-0.01) | Kernel=linear  C=216  Gamma= None |  | %83 |
| MLP | hidden\_layer\_sizes=100 activation='relu' solver='adam'  alpha=0.0001 | %79.2 | hidden\_layer\_sizes=[(10,30,10),(20,)]  activation=[tanh, relu]  solver=['sgd', 'adam']  alpha=[0.0001,0.001,0.01] | hidden\_layer\_sizes=(20,)  activation=’relu' solver='adam'  alpha=0.001 |  | %82.4 |

1. **Conclusion**

In this study, author-artwork matching was performed using the columns published on the website of a newspaper. For this purpose, firstly, the dataset was organised with basic machine learning libraries and some data preprocessing techniques to make it suitable for classification. Then, the Zemberek NLP library and feature extraction methods (TF-IDF and Word Bag) were used in order to obtain maximum efficiency from the dataset. SVM and MLP algorithms were used in the classification phase. Hyperparameter analysis was used to improve the classification success rate. As a result of the parameter analysis, the highest success rate was obtained by using the SVM algorithm with 83%.

**References**

[1] Talan, T. ve Aktürk, C., 2021, Bilgisayar Bilimlerinde Teorik Ve Uygulamalı Araştırmalar, p.

[2] Oğuzlar, A., 2003, Veri ön işleme, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi (21).

[3] Famili, A., Shen, W.-M., Weber, R. ve Simoudis, E., 1997, Data preprocessing and intelligent data analysis, Intelligent data analysis, 1 (1), 3-23.

[4] Dondurmacı, G. A. ve Çınar, A., 2014, Finans sektöründe veri madenciliği uygulaması, Akademik Sosyal Araştırmalar Dergisi, 2 (1), 258-271.

[5] Yavuz, S. ve Deveci, M., 2012, İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağin Performansina Etkisi, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi (40), 167-187.

[6] Gültepe, Y., 2019, Makine öğrenmesi algoritmaları ile hava kirliliği tahmini üzerine karşılaştırmalı bir değerlendirme, Avrupa Bilim ve Teknoloji Dergisi (16), 8-15.

[7] Erdal, H. ve Yapraklı, T. Ş., 2016, Firma başarısızlığı tahminlemesi: Makine öğrenmesine dayalı bir uygulama, Bilişim Teknolojileri Dergisi, 9 (1), 21.

[8] Akın, A. A. ve Akın, M. D., 2007, Zemberek, an open source NLP framework for Turkic languages, Structure, 10 (2007), 1-5.

[9] Gündoğdu, Ö. E. ve Duru, N., 2016, Türkçe Metin Özetlemede Kullanılan Yöntemler, 18. Akademik Bilişim Konferansı-AB'16.

[10] Ekin, M. F., 2020, Türkçe morfolojik analiz için yeni bir yöntem, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü.

[11] Uludogan, G., Özçelik, R., Parlar, S., Ercan, G. ve Yıldız, O. T., 2017, Türkçe Dogal DilIsleme için Arayüzler User Interfaces for Turkish Natural Language Processing, IEEE.

[12] Taşkıran, S. F., 2021, Doğal dil işleme ile akademik metinlerin kümelenmesi, Konya Teknik Üniversitesi.

[13] Christian, H., Agus, M. P. ve Suhartono, D., 2016, Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF), ComTech: Computer, Mathematics and Engineering Applications, 7 (4), 285-294.

[14] Ataseven, B., 2013, Yapay sinir ağlari ile öngörü modellemesi, Öneri Dergisi, 10 (39), 101-115.

1. \* Corresponding author. *e-mail address: ibrahimdgn023@gmail.com* [↑](#footnote-ref-1)