Sakarya University Journal of Computer and Information Sciences
http://saucis.sakarya.edu.tr/

DOI: 10.35377/saucis...1207742

RESEARCH ARTICLE

# The Effects of Preprocessing on Turkish and English News Data

**Bekir Parlak**[1]

[1]Amasya University, Türkiye

**Corresponding author:**
Bekir Parlak, Amasya University
**E-mail address:**
bekir.parlak@amasya.edu.tr

**Submitted:**21 Nov 2022
**Revised:** 27 March 2023
**Accepted:** 30 March 2023
**Published Online:** 30 April 2023

**Citation:** Parlak,B. (2023).
The effects of preprocessing on Turkish and English News Data.
*Sakarya University Journal of Computer and Information Sciences* 6 (1)
https://doi.org/10.35377/saucis...1207742

## ABSTRACT

In a standard text classification (TC) study, preprocessing is one of the key components to improve performance. This study aims to look at how preprocessing effects TC according to news text, text language, and feature selection. All potential combinations of commonly used preprocessing techniques were compared on one domain, namely news data, and two different news datasets for this aim. Preprocessing technique contributions to classification performance at multiple feature sizes, possible interconnections among these techniques, and technique dependency on corresponding languages were all evaluated in this way. The effect of two important preprocessing techniques on two different common news datasets was examined. While the highest performance for the Turkish dataset is a 0.781 F1 score, the highest performance for the English dataset is a 0.980 F1 score.

**Keywords:** Feature selection, news data, preprocessing, text classification

## 1. Introduction

TC is one of the most challenging study subjects due to the requirement to organize and classify an increasing number of digital text documents globally. TC has been efficiently used in many different fields.

Preprocessing, feature extraction, feature weighting, feature selection, and classification are all steps in a traditional TC system. Stop-word removal, stemming, tokenization, and lowercase conversion, are common tasks in the preprocessing stage. In most cases, the feature extraction step uses the vector space model[1] which employs the bag-of-words technique[2]. Filter methods such as document frequency[3], information gain[4], Gini index[5], Normalized Difference Measure(NDM)[6], Max-Min Ratio(MMR)[7], Extensive Feature Selector[8] and Class-index Corpus-index Measure(CiCi)[9] are used to the feature selection step for TC domain. As a final, the classification step employs successful and well-known classification methods, such as nave Bayesian classifiers, artificial neural networks, decision trees, support vector machines, and among others.

While it has been established that feature extraction, feature weighting, feature selection, and classification method have a significant effect on the performance of TC, the preprocessing step may also have a significant impact. Stemming, stop-word removal, alphabetic tokenization and lowercase conversion are commonly used in text categorization research without thoroughly analyzing their contributions to classification accuracy.

One investigation into the categorization of Turkish news data, Kılınç et al. [10] created a new dataset called TTC-3600 that may be extensively used in TC research of Turkish news and article content. On TTC-3600, different successful classifiers in the TC domain and successful feature selection methods are evaluated. The experimental studies show that the combination of the Random Forest (RF) classifier and filter-based feature selection method achieves the best performance in all comparisons performed after pre-processing techniques and feature selection steps. In another study[11], a TC study was carried out on the TTC-3600 dataset using Convolutional Neural Networks (CNN) and Word2Vec method and compared with the previous study using the same dataset. In the study, two different CNNs were trained and tested on the raw and stemmed versions of the TTC-3600 with the Zemberek library. CNN and Word2Vec methods showed more performance than classical statistical and machine learning algorithms. Yıldırım et al.[12] were compared traditional bag-of-words approach and neural network based new representation approaches in terms of TC. In this study, it seems that the traditional methods of effective feature selection are still at a level to compete with the new generation word embeddings approach. Experiments are reported by diversifying in terms of these two approaches and successful TC architecture for Turkish is discussed in detail. Safali et al.[13] classified academic studies based on deep learning using the Doc2vec word embeddings method. For the training, 7 different symposiums broadcasting in Turkey were selected. During the classification process, it was ensured that the studies were classified into 9 different categories by using recurrent neural networks (RNNs) and LSTM architectures. Köksal[14] conducted experiments with the TTC-4900 dataset. This dataset is comparable to TTC-3600. The TTC-4900 dataset contains 4900 news documents and 700 examples of Turkish and English news data from seven different classes. The study made extensive use of data correction. Stop words in Turkish and English are then removed. Finally, the process of root separation (lemmatization) is used. The F1 score increased when the original data was corrected, while it decreased when lemmatized. As a result, the original dataset received % 90 F1 score, but the F1 score increased to % 91.77, after correcting the data without using lemmatizing.

One investigation into the categorization of English news data, Dadgar et al.[15] objects to categorize news. It was proposed using Term Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machine (SVM). The proposed method consists of three steps: 1) data preprocessing, 2) TF-IDF feature weighting, and 3) SVM classification. The proposed method was tested using two datasets. The classification performances for the BBC and 20Newsgroup datasets were 97.84 and 94.93 percent, respectively. When compared to other classification methods, these are very desirable results. Haryanto et al.[16] can enhance the effectiveness of TC using Chi-square feature selection and SVM on preprocessing data results, including lemmatization and stemming. In another study[17], TC is carried out using a new bi-gram approach instead of the unigram approach to construct a feature vector. The proposed method makes significant contributions to the field of TC. In a study[18], researchers focused on identifying sentence level negations in news articles. This work makes use of online news articles from BBC News. Machine Learning Algorithms such as SVM and Nave Bayes are used to analyze the results. SVM has an accuracy of 96.46 percent, while Naive Bayes has an accuracy of 94.16 percent.

The effect of two techniques, which are the most important pre-processing steps, on text classification performance is analyzed in detail in this study. These techniques are stemming and stopwords removal. These two techniques have a serious impact in terms of dimension reduction. Because after these techniques are applied, the feature size decreases as unnecessary and similar words are discarded. In addition, these techniques were applied to Turkish and English datasets to examine how they changed performance on the basis of languages.

The remainder of this research is structured as follows. In section 2, materials and methods are explained. These are datasets, preprocessing, feature selection and representation, and classification algorithms. In section 3, we described experimental settings consisting of classifiers, and accuracy analysis. Finally, a conclusion is given in section 4.

## 2. Materials and Methods

In this section, the datasets, preprocessing techniques, feature selection methods, feature weighting and pattern classifiers used in the experiments are explained in detail. In Section 2.1, datasets are presented in detail. Preprocessing techniques are explained in Section 2.2. The feature selection techniques used in this study are explained in detail in Section 2.3. Information about feature representation and feature weighting is given in Section 2.4. Finally, the classification algorithms used in this study are given in Section 2.5. The flowchart of the study is shown in Figure 1.

### 2.1 Datasets

Preprocessing techniques are assessed in two different datasets, and on just one topic, namely news. English is a non-agglutinative language, despite Turkish being one of the world's most often spoken agglutinative languages. The number of documents within the same categories is kept essentially constant to allow for an impartial review. The Turkish news dataset includes 300 training and 300 test samples for each class. The English news datasets include 500 training samples and 500 test examples for each class. The number of features reaches hundreds of thousands in studies in the field of text classification. For this reason, it is sufficient that the training-test rate is 50%-50%. In addition, very high scores are obtained when the training data is above 50%. Tables 2-3 summarize the class distributions for news datasets. Two news datasets are balanced. The preprocessing tasks for the multi-class classification problem are evaluated using the news datasets. The first dataset is TTC-3600[10]. Being user-friendly and well-documented is the most crucial aspect of this dataset, which may be extensively employed in

TC studies pertaining to Turkish news and articles. The dataset contains 3600 documents in total, 600 of which are news stories or texts from six different categories. These articles were gathered from six reputable news portals and agencies. The second dataset is 20Newsgroups. The newsgroups dataset is evenly distributed, with equal documents in all classes. The number of documents in each class of the datasets are presented in Table 1 and Table 2.

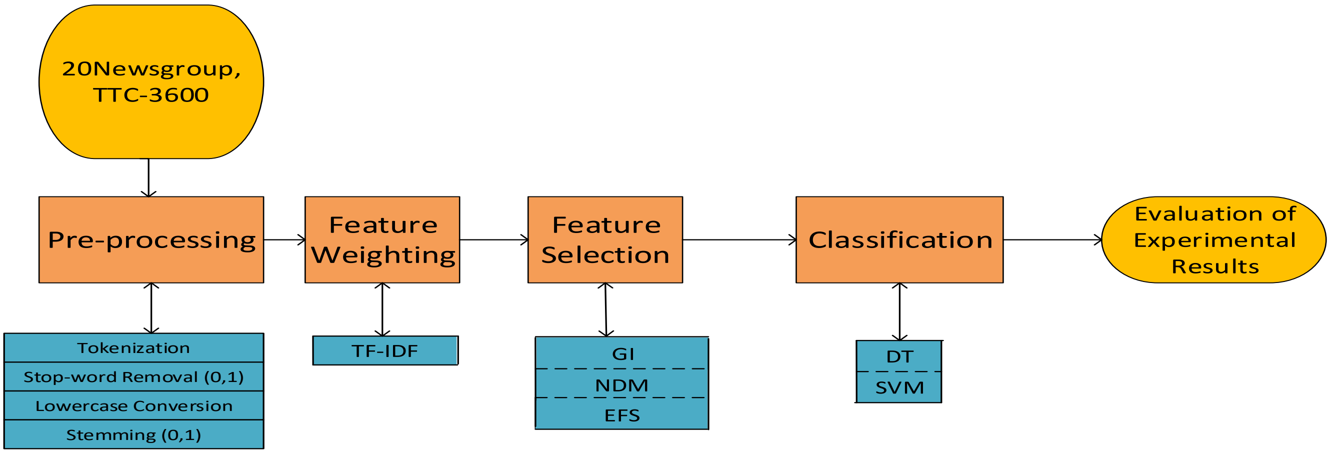

Figure 1 Flowchart of the study

Table 1 TTC-3600 Dataset

| Categories | Total | Train | Test |
|---|---|---|---|
| Culture | 600 | 300 | 300 |
| Economy | 600 | 300 | 300 |
| Health | 600 | 300 | 300 |
| Policy | 600 | 300 | 300 |
| Sport | 600 | 300 | 300 |
| Technology | 600 | 300 | 300 |
| Total | 3600 | 1800 | 1800 |

Table 2 20Newsgroup

| Categories | Total | Train | Test |
|---|---|---|---|
| alt.atheism | 1000 | 500 | 500 |
| comp.graphics | 1000 | 500 | 500 |
| comp.os.ms-windows.misc | 1000 | 500 | 500 |
| comp.sys.ibm.pc.hardware | 1000 | 500 | 500 |
| comp.sys.mac.hardware | 1000 | 500 | 500 |
| comp.windows.x | 1000 | 500 | 500 |
| misc.forsale | 1000 | 500 | 500 |
| rec.autos | 1000 | 500 | 500 |
| rec.motorcycles | 1000 | 500 | 500 |
| rec.sport.baseball | 1000 | 500 | 500 |
| Total | 10000 | 5000 | 5000 |

### 2.2 Pre-processing techniques

Within the scope of this study, four common preprocessing steps of TC are considered: stop-word removal, stemming, lowercase conversion, and tokenization[19]. Of these techniques, lowercase conversion and tokenization do not vary with the language of the dataset, while stop-word removal and stemming techniques vary with the language of the dataset. Because the method of finding the stem of the words in each language and the stop-word list is different. However, tokenization and lowercase conversion are the same in every language, as no structural processing is applied to the word. Stemming algorithms are customized to the language according to the study. Among the various approaches, the fixed-prefix algorithm [20] is a simple but highly effective stemming technique for Turkish. However, the stemming algorithm, namely Porter-stemmer introduced in a study[21] is widely used by English researchers. In this study, Zemberek was used for the Turkish dataset and Porter-stemmer was used for the English dataset. Zemberek is an open-source natural language processing library that you can use for Turkish languages, developed by Ahmet A. Akın using the Java programming language. By using this library, besides finding the stem words, spelling, checking whether a word is Turkish, correcting spelling mistakes etc. can be done.

There are some rules specific to the Turkish language in the Zemberek library. Today, there is no algorithm and library that works more accurately than the Zemberek library for Turkish in detecting word stem. It is aimed to find the stem of the word by removing the suffixes such as constructional affixes and inflectional affixes.

### 2.3 Feature selection methods

In this study, it is employed three different feature selection methods. These methods are Gini Index, Normalized Difference Measure, Extensive Feature Selector, Odds Ratio, and Chi-Square. While GI, NDM, and EFS are global methods, OR and CHI2 are local. So, we utilized global and local methods in the experiments. All notations used in these methods are shown in Table 3.

Table 3 Notations

| Notation | Meaning |
|---|---|
| $P(t\|C_j)$ | Probability of term t when class $C_j$ exists |
| $P(\bar{t}\|C_j)$ | Probability of absence of term t when class $C_j$ exists |
| $P(t\|\bar{C}_j)$ | Probability of term t when class $C_j$ does not exists |
| $P(\bar{t}\|\bar{C}_j)$ | Probability of absence of term t when class $C_j$ does not exists |
| $P(C_j\|t)$ | Probability of class $C_j$ when term t exists |
| $P(\bar{C}_j\|t)$ | Probability of absence of class $C_j$ when term t exists |
| $P(C_j\|\bar{t})$ | Probability of class $C_j$ when term t does not exist |
| $P(\bar{C}_j\|\bar{t})$ | Probability of absence of class $C_j$ when term t does not exist |

#### 2.3.1 Gini Index (GI)

GI is a successful technique for the TC domain. A novel measure of the Gini index is created in order to fit TC. Following an examination of the advantages and disadvantages of the current text feature selection measure functions, the GI formula is as follows:

$$\boldsymbol{GI(t) = \sum_{i=1}^{M} P(t|C_j)^2 \cdot P(C_j|t)^2} \tag{1}$$

#### 2.3.2 Normalized Difference Measure (NDM)

The NDM[6] technique, which takes into account relative document frequencies. The Balanced Accuracy Measure has been improved with the NDM method. The NDM algorithm computes as follows:

$$\boldsymbol{NDM(t) = \sum_{i=1}^{M} \frac{|P(t|\ C_j) - P(t|\bar{C}_j)|}{\min\left(P(t|\ C_j), P(t|\bar{C}_j)\right)}} \tag{2}$$

#### 2.3.3 Extensive Feature Selector (EFS)

EFS is a successful technique for the TC domain. A novel measure of the EFS is created in order to fit TC. Following an examination of the advantages and disadvantages of the current text feature selection measure functions, the EFS formula is as follows:

$$\boldsymbol{EFS(t) = \sum_{j=1}^{M} \left(\frac{P(t|C_j)}{P(\bar{t}|C_j) + P(t|\bar{C}_j) + 1}\right) \cdot \left(\frac{P(C_j|t)}{P(\bar{C}_j|t) + P(C_j|\bar{t}) + 1}\right)} \tag{3}$$

### 2.4 Feature Representation and Weighting

In general, it has been evaluated text documents as a bag of words (BoW) approach in machine learning classifiers. In an enhanced form of BoW known as the Vector Space Model (VSM), each document is represented as a vector, with each dimension denoting a distinct term (word) or feature. A term's value in the vector changes from zero to non-zero if it appears in the text. The objective, as viewed from a TC perspective, is to create vectors with features for each class using training documents. Term weighting is a crucial stage in VSM, and there are three main factors that influence how important a term is in a document. These factors are document length normalization, Term frequency factor ($TF$), and the inverse document frequency factor ($IDF$).

## 2.5 Classification algorithms

The goal of TC is the classification of unclassified documents into predetermined classes. Machine learning classifiers for TC are well-documented in the literature. In this study, we employed two successful pattern classifiers, namely Decision Tree (DT) and Support Vector Machine (SVM). The following section provides illustrations of the specific details about each classifier that was chosen.

### 2.5.1 J48 Decision Tree (DT)

DT learning is a supervised machine learning classifier that uses a DT to classify an input document and determine its category. Internal nodes in the DT represent dataset attributes, leaves represent classification labels, and branches represent attribute values, respectively. J48 grows the DT using a divide-and-conquer approach. J48 is particularly successful in the field of TC.

### 2.5.2 Support Vector Machine (SVM)

SVM, is a successful classifier based on statistical information theory and structural risk minimization. SVM classifier is split into linear and nonlinear SVM algorithms. An infinite number of hyper-planes are formed to divide the data, and the hyper-plane with the highest margin is chosen from all of them in the linear SVM algorithm. When classes cannot be distinguished linearly and data must be translated into a higher dimensional space, nonlinear SVM is utilized. The data can also be separated linearly as a result. High accuracy and resistance to over-fitting via structural risk minimization by utilizing a regularization parameter are the major benefits of SVM.

## 3. Experimental Study

In this study, three different filter feature selection methods are used to evaluate according to the performance applied on 20Newsgroups and TTC-3600 datasets. DT and SVM classifiers were fed different sizes of features chosen by each feature selection approach. Additionally, the translation of documents into a term-document matrix has been carried out with the Java programming language. Classification has been carried out using the Weka software tool. The total feature size varies according to the experimental parameters. The application of the stemming algorithm and the removal of stop-words change the feature size. If these processes are not applied for two datasets, feature size increases. However, when the stemming algorithm is not applied in the TTC-3600 dataset, the number of features has increased significantly. The feature numbers are given in Table 4-5 according to the datasets and situations. The resulting F1-Scores are shown in Tables 6-9 for TTC-3600 and 20Newsgroup datasets where the highest performance is underlined and bold.

The highest score for the TTC-3600 dataset was obtained with the EFS method, SVM classifier and 1000 feature size. Also, this score was obtained without a stemming algorithm. Similarly, the highest score for the DT classifier was obtained with the NDM method and 1000 feature size and not applying the stemming algorithm. From here, it has been seen that Zemberek[22], the Turkish stemming algorithm, does not make a serious contribution to the performance for the TTC-3600 dataset. In general, performance increases as the feature size increases for the TTC-3600 dataset.

Table 4 Total number of features for TTC-3600

| Situations | Total # of features |
|---|---|
| Stemming=1, Stopwords=1 | 19605 |
| Stemming=1, Stopwords=0 | 19672 |
| Stemming=0, Stopwords=1 | 62197 |
| Stemming=0, Stopwords=0 | 62402 |

Table 5 Total number of features for 20Newsgroup

| Situations | Total # of features |
|---|---|
| Stemming=1, Stopwords=1 | 39912 |
| Stemming=1, Stopwords=0 | 40166 |
| Stemming=0, Stopwords=1 | 49938 |
| Stemming=0, Stopwords=0 | 50451 |

Table 6 F1 scores from TTC-3600 dataset with Decision Tree

| **Feature Selection Methods** | **50** | **100** | **300** | **500** | **1000** |
|---|---|---|---|---|---|
| (Stemming=1,Stopwords=1) | | | | | |
| GI | 0.636 | 0.609 | 0.683 | 0.684 | 0.701 |
| NDM | 0.410 | 0.570 | 0.687 | 0.692 | 0.689 |
| EFS | 0.571 | 0.626 | 0.670 | 0.691 | 0.702 |
| (Stemming=1,Stopwords=0) | | | | | |
| GI | 0.581 | 0.624 | 0.681 | 0.683 | 0.699 |
| NDM | 0.410 | 0.570 | 0.680 | 0.689 | 0.688 |
| EFS | 0.546 | 0.581 | 0.657 | 0.687 | 0.689 |
| (Stemming=0,Stopwords=1) | | | | | |
| GI | 0.601 | 0.644 | 0.682 | 0.707 | 0.700 |
| NDM | 0.526 | 0.569 | 0.677 | 0.659 | 0.709 |
| EFS | 0.572 | 0.627 | 0.673 | 0.694 | 0.699 |
| (Stemming=0,Stopwords=0) | | | | | |
| GI | 0.519 | 0.630 | 0.655 | 0.687 | 0.703 |
| NDM | 0.526 | 0.569 | 0.677 | 0.678 | 0.699 |
| EFS | 0.506 | 0.589 | 0.639 | 0.687 | 0.698 |

Table 7 F1 scores from TTC-3600 dataset with Support Vector Machine

| **Feature Selection Methods** | **50** | **100** | **300** | **500** | **1000** |
|---|---|---|---|---|---|
| (Stemming=1,Stopwords=1) | | | | | |
| GI | 0.671 | 0.662 | 0.728 | 0.750 | 0.762 |
| NDM | 0.469 | 0.614 | 0.716 | 0.717 | 0.738 |
| EFS | 0.535 | 0.662 | 0.720 | 0.735 | 0.762 |
| (Stemming=1,Stopwords=0) | | | | | |
| GI | 0.634 | 0.660 | 0.727 | 0.758 | 0.765 |
| NDM | 0.469 | 0.614 | 0.716 | 0.728 | 0.749 |
| EFS | 0.595 | 0.647 | 0.733 | 0.754 | 0.764 |
| (Stemming=0,Stopwords=1) | | | | | |
| GI | 0.612 | 0.681 | 0.723 | 0.733 | 0.764 |
| NDM | 0.555 | 0.636 | 0.710 | 0.718 | 0.732 |
| EFS | 0.599 | 0.652 | 0.710 | 0.727 | 0.781 |
| (Stemming=0,Stopwords=0) | | | | | |
| GI | 0.560 | 0.665 | 0.729 | 0.743 | 0.775 |
| NDM | 0.555 | 0.636 | 0.710 | 0.721 | 0.744 |
| EFS | 0.563 | 0.627 | 0.717 | 0.727 | 0.768 |

Table 8 F1 scores from 20Newsgroup dataset with Decision Tree

| **Feature Selection Methods** | **50** | **100** | **300** | **500** | **1000** |
|---|---|---|---|---|---|
| (Stemming=1,Stopwords=1) | | | | | |
| GI | 0.977 | 0.975 | 0.947 | 0.947 | 0.972 |
| NDM | 0.689 | 0.906 | 0.973 | 0.972 | 0.972 |
| EFS | 0.977 | 0.948 | 0.946 | 0.945 | 0.973 |
| (Stemming=1,Stopwords=0) | | | | | |
| GI | 0.976 | 0.975 | 0.946 | 0.946 | 0.970 |
| NDM | 0.689 | 0.906 | 0.979 | 0.979 | 0.980 |
| EFS | 0.976 | 0.976 | 0.946 | 0.945 | 0.945 |
| (Stemming=0,Stopwords=1) | | | | | |
| GI | 0.845 | 0.913 | 0.951 | 0.955 | 0.949 |
| NDM | 0.353 | 0.581 | 0.720 | 0.920 | 0.931 |
| EFS | 0.846 | 0.913 | 0.928 | 0.951 | 0.951 |
| (Stemming=0,Stopwords=0) | | | | | |
| GI | 0.899 | 0.920 | 0.921 | 0.952 | 0.949 |
| NDM | 0.353 | 0.581 | 0.874 | 0.910 | 0.921 |
| EFS | 0.878 | 0.888 | 0.915 | 0.923 | 0.953 |

Table 9 F1 scores from 20Newsgroup dataset with Support Vector Machine

| Feature Selection Methods | 50 | 100 | 300 | 500 | 1000 |
|---|---|---|---|---|---|
| (Stemming=1,Stopwords=1) | | | | | |
| GI | 0.970 | 0.968 | 0.965 | 0.960 | 0.960 |
| NDM | 0.701 | 0.898 | 0.976 | 0.973 | 0.966 |
| EFS | 0.971 | 0.969 | 0.963 | 0.960 | 0.961 |
| (Stemming=1,Stopwords=0) | | | | | |
| GI | 0.969 | 0.971 | 0.962 | 0.963 | 0.962 |
| NDM | 0.701 | 0.898 | 0.973 | 0.974 | 0.967 |
| EFS | 0.972 | 0.970 | 0.964 | 0.963 | 0.963 |
| (Stemming=0,Stopwords=1) | | | | | |
| GI | 0.839 | 0.922 | 0.941 | 0.927 | 0.902 |
| NDM | 0.411 | 0.604 | 0.757 | 0.923 | 0.931 |
| EFS | 0.844 | 0.927 | 0.907 | 0.912 | 0.904 |
| (Stemming=0,Stopwords=0) | | | | | |
| GI | 0.901 | 0.925 | 0.910 | 0.913 | 0.912 |
| NDM | 0.411 | 0.604 | 0.886 | 0.923 | 0.930 |
| EFS | 0.902 | 0.902 | 0.892 | 0.884 | 0.912 |

The highest score for the 20Newsgroup dataset [23] was obtained with the NDM method, DT classifier and 1000 feature size. In addition, this score was obtained by performing a stemming algorithm. Similarly, the highest score for the DT classifier was obtained by applying the NDM method and the feature size of 300, the removal of stopwords, and applying the stemming algorithm. From here, it has been seen that Porter, the English stemming algorithm, contributes to the performance [24]. In general, performance improves as the feature size increases for the 20Newsgroup dataset. In some cases, performance decreases as the size increases.

## 4. Results and Discussion

Experimental results have shown which method produces the highest and lowest performance. In addition, the effect of preprocessing methods on performance has been analyzed in detail. Both preprocessing techniques used significantly change the number of features. In terms of size, it had a positive contribution to performance in both datasets in general. In addition, when the stemming algorithm is not applied for Turkish, the feature size increases more than the English dataset. It was observed that applying the preprocessing technique did not have a positive effect on the performance for the Turkish dataset. However, preprocessing techniques contributed significantly to the performance for the English dataset. For this reason, the development of more effective stemming algorithms for Turkish will be a research topic for researchers in this field.

## 5. Conclusions

This study looked closely at how commonly used preprocessing tasks affected TC in just one domain and two different languages. The examination was conducted utilizing every possible combination of the preprocessing tasks while considering different factors including accuracy, language, and dimension reduction. Extensive experimental investigation showed that the right preprocessing task combinations, depending on the language, may significantly improve classification accuracy, whereas the wrong preprocessing task combinations may reduce classification accuracy. As a result, the preprocessing stage in the TC process is just as crucial as the other TC steps.

The two datasets examined in this work each have unique characteristics in terms of language, class distribution, and a number of classes. Hence the conclusions drawn from this study may also apply to other text collections.

## References

[1] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing". *Communications of the ACM*, 1975. 18(11): p. 613-620.

[2] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features". in *European conference on machine learning*. 1998. Springer.

[3] Y. Yang, and J.O. Pedersen. "A comparative study on feature selection in text categorization." in *ICML*. 1997.

[4] C. Lee, and G.G. Lee," Information gain and divergence-based feature selection for machine learning-based text categorization." *Information processing & management*, 2006. 42(1): p. 155-165.

[5] S.R. Singh, H.A. Murthy, and T.A. Gonsalves, "Feature Selection for Text Classification Based on Gini Coefficient of

Inequality. "*Fsdm*, 2010. 10: p. 76-85.

[6] A. Rehman, K. Javed, and H.A. Babri, "Feature selection based on a normalized difference measure for text classification." *Information Processing & Management*, 2017. 53(2): p. 473-489.

[7] A. Rehman, et al., "Selection of the most relevant terms based on a max-min ratio metric for text classification." *Expert Systems with Applications*, 2018. 114: p. 78-96.

[8] Parlak, B. and A.K. Uysal, A novel filter feature selection method for text classification: Extensive Feature Selector. Journal of Information Science, 2021: p. 0165551521991037.

[9] B. Parlak, "Class-index corpus-index measure: A novel feature selection method for imbalanced text data." C*oncurrency and Computation: Practice and Experience*, 2022: p. e7140.

[10] D. Kilinc, et al., "TTC-3600: A new benchmark dataset for Turkish text categorization." *Journal of InformationScience*, 2017. 43(2): p. 174-185.

[11] A. Çiğdem. and A. Çırak, "Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması." *Bilişim Teknolojileri Dergisi*, 2019. 12(3): p. 219-228.

[12] S. Yıldırım, and T. Yıldız, "Türkçe için karşılaştırmalı metin sınıflandırma analizi. "*Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 2018. 24(5): p. 879-886.

[13] Y. Safali, et al. "Deep learning based classification using academic studies in doc2vec model". in 2*019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. 2019. IEEE.

[14] Ö. Köksal, "Tuning the Turkish Text Classification Process Using Supervised Machine Learning-based Algorithms". in *2020 International Conference on INnovations in Intelligent SysTems and Applications* (INISTA). 2020. IEEE.

[15] S.M.H. Dadgar, M.S. Araghi, and M.M. Farahani. "A novel text mining approach based on TF-IDF and Support Vector Machine for news classification." in *2016 IEEE International Conference on Engineering and Technology (ICETECH).* 2016. IEEE.

[16] A.W. Haryanto, and E.K. Mawardi. "Influence of word normalization and chi-squared feature selection on support vector machine (svm) text classification." in *2018 International Seminar on Application for Technology of Information and Communication.* 2018. IEEE.

[17] F. Elghannam, "Text representation and classification based on bi-gram alphabet." J*ournal of King Saud University-Computer and Information Sciences,* 2021. 33(2): p. 235-242.

[18] V.S. Shirsat, R.S. Jagdale, and S.N. Deshmukh, "Sentence level sentiment identification and calculation from news articles using machine learning techniques," in *Computing, Communication and Signal Processing.* 2019, Springer. p. 371-376.

[19] A.K. Uysal, and S. Gunal, "The impact of preprocessing on text classification." *Information Processing & Management*, 2014. 50(1): p. 104-112.

[20] D. Torunoğlu, et al. "Analysis of preprocessing methods on classification of Turkish texts." In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2011. p. 112-117.

[21] M.F. Porter, "An algorithm for suffix stripping." *Program*, 1980. 14(3): p. 130-137.

[22] A. Akın, M. D. Zemberek, "an open source NLP framework for Turkic languages". *Structure*, 2007, 10.2007: 1-5.

[23] B. Parlak, and A.K. Uysal, "The effects of globalization techniques on feature selection for text classification." *Journal of Information Science*, 2021, 47(6), 727-739.

[24] B. Parlak and A.K. Uysal, "On classification of abstracts obtained from medical journals." *Journal of Information Science*, 2020, 46(5), 648-663.

**Acknowledgments**

This work was not supported by any fund or project.

**Conflict of Interest Notice**

The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

**Availability of Data and Material**

Not applicable.

**Plagiarism Statement**

This article has been scanned by iThenticate ™.