CSE 454 Data Mining
Final Project Report

Ahmed Semih Özmekik
171044039

January 24, 2021

Abstract

Detailed explanation of design choices along with the experimental results in the homework.

1 Project Definition

Our main goal in the project was to use many data mining algorithms in the literature that can be used at the point where we will specialize by applying many preprocessing, postprocessing, methods related to data mining.

Based on this, we first determined a paper. [1] Then, we focused on improving the results obtained there by examining the experimental study in the paper. This project was firstly an implementation of a paper, and then by doing different studies on the dataset suggested in this paper, it was aimed and achieved better scores than the scores obtained in this paper.

The subject is text classification. There are many different and successful methods in the current literature on this subject. We present the results of 8 different methods. For each method, there are 6 different preprocessing parameters, 2 different feature vector method parameters, and 5 different feature number definition parameters for the feature selection method. In other words, 8x2x5 = 80 different results were obtained for each method, hence 80x8 = 640 different results obtained in total.

For these 480 different results, by making some comparisons between them, the results of all methods and independent variables in the experiment were analyzed, and the dependent variable f1 score was observed. As a result, all data were recorded and the most successful parameter selection was determined.

2 Text Classification

The number of electronic documents produced as a result of the transition to the electronic world is increasing day by day. Manual processing or classification of these electronic documents in text format, whose number is rapidly growing, has become almost impossible today. Today, text classification is carried out by machine learning or deep learning methods. Text classification, sorting the content of the text according to the specified categories is the process.[2, 3]

There are many noise words in a text that are not specific to that text and do not represent an attribute for the classification problem. [4] That's why preprocessing is important in the TC problem.

After applying the necessary pre-operations to the text, a feature must be extracted from the text, in other words, the text must be converted into a feature vector. For this, there are many vectorization studies such as word2vec, bag of words.

After obtaining this vector, it is a classification method that will be applied. For example, there are many statistics based methods such as naive bayes, support vector machine.

3 Dataset

The name of the dataset is TTC-3600.

The dataset includes a total of 3600 documents, 600 of them from each class (economy, culture-arts, health, politics, sports and technology), all collected from well-known and known news portals.

In the reference study, only 3 variances were produced from this dataset, and there were 4 different datasets in total, including the raw dataset.

The first of these is the raw dataset that we will call Original-DS throughout this study, on which no preprocess has been applied (except for the correction of html, css tags during the data set collection stage).

The second is the dataset, named F5-DS, processed using FPS-5 (mentioned in the preprocessing section) as stemmer. FPS-7 was used as stemmer in the third one, F7-DS, and Zemberek was used as stemmer in the last one, Zemb-DS.

Here, we further diversified our experiment based on this study. By processing the raw dataset in different ways, we had 8 different datasets in total. Now let's talk in detail about preprocessing processes that we implement.

4 Preprocessing

Preprocessing is the most important process of TC subject. At this stage, we have applied several methods on our data set, usually based on removal, that is, to remove noise on the data.

First of all, we extracted the numbers from the whole document. This work was not done in the article. Then, we performed a normalization by converting all words to lower cases, and operations such as removing punctuation marks, separators, operators, or meaningless characters were performed. Finally, all words were tokenized. These were common pre-processing operations across all datasets (including the raw set).

Then here, we first created 3 more datasets by applying 3 different stemming methods. These are the processes using the FPS-5, FPS-7, and Zemberek method.

The FPS method is a simple stemming method where only the first 'n' letter of the word is kept. [5] In FPS-5, only the first 5 letters are kept, and in FPS-7, only the first 7 letters are kept and other letters are deleted.

Zemberek, on the other hand, is a Turkish dictionary-based tool for stemming. [6] Here, too, we see that besides the simple assumption in FPS, a more in-depth and accurate root separation is applied. But we will see how it affects the overall result with the experiments we have done.

In the reference study, words called stopwords, which do not represent any meaning and feature in the content of the text, were removed from the whole data set, such as conjunctions and prepositions.

In our research, we did not accept this assumption, and we divided our data set, which was 4 in total, into 2, with 3 different datasets created by raw and the stemming processes mentioned above, one with stopwords removed and no stopwords removed, and in total we had 8 different preprocess product datasets.

Below, you can examine these datasets, along with their names, in more detail.

Dataset name	Stop word filtering	Stemmer
Original-DS	No	No stemmer
F5-DS	No	FPS-5
F7-DS	No	FPS-7
Zemb-DS	No	Zemberek
OriginalSW-DS	Yes	No Stemmer
F5SW-DS	Yes	FPS-5
F7SW-DS	Yes	FPS-7
ZembSW-DS	Yes	Zemberek

5 Postprocessing

5.1 Feature Extraction

After preprocessing the datasets, we prepared independent variables for all our experiments, where another processing parameters can be specified.

The first category represents the different methods that can be applied in extracting a feature vector from the texts we have. We chose two approaches in the literature to use in our experiments: Bag of words and TFIDF vectorization methods.

These methods are methods that infer a feature vector from a text that represents that text. Bag of words method, as its name also hints, defines a feature vector by approaching texts as a word bag, losing the spatial data of the words, that is, the sequence information, using only the word histogram in that text. Compared to being a simple method, it has yielded successful results in the literature.

TFIDF (term frequency-inverse document frequency), on the other hand, is a relatively slightly more complex numerical statistic that aims to reflect how important a word is to a document in a collection or collection.

In both feature extraction processes, the maximum number of feature can be obtained is limited to 8000. The reference study recorded the highest feature number in the raw dataset, 7508.

5.2 Feature Selection

After the feature is obtained from the texts, another important independent variable is the parameter that specifies 'n' features selected by chi2 feature selection method among these features. The user can give the desired number and parameter. In the context of our experiments, we conducted this experiment through discrete values, namely 500, 1000, 2000, 5000 or 'all' as the feature number, with options to select all features.

These two parameters, the method to be applied for feature extraction and the feature number to be selected in feature selection, indicate the last parameters in the data process. From now on, the only parameter that can be changed is the classification method to be applied on the ready data. Let's examine them now.

6 Methods

6.1 Naive Bayes

The Naive Bayes classifier is based on Bayes' theorem. It is a lazy learning algorithm, it can also work on unstable datasets. The way the algorithm works calculates the probability of each state for an element and classifies it according to the one with the highest probability value. With a little training data, he can do very successful jobs. If a value in the test set has an unobservable value in the training set, it gives 0 as a probability value, which means it cannot predict. This condition is commonly known as Zero Frequency. Correction techniques can be used to resolve this situation. One of the simplest correction techniques is known as Laplace estimation. Examples of usage areas are real-time prediction, multi-class prediction, text classification, spam filtering, sentiment analysis and suggestion systems.

6.2 Random Forest

Random Forest is one of the popular machine learning models because it can be applied to both regression and classification problems, giving good results without hyperparameter estimation. To understand the random forest, it is necessary to first understand the decision trees, which is the basic blog of this model. We have devoted the 3rd lesson to this subject, a successful decision tree can be compared to people who ask questions and make accurate predictions that will increase their knowledge gain in daily life.

However, one of the biggest problems of decision trees, which is one of the traditional methods, is over-learning-overfitting. In order to solve this problem, the random forest model randomly selects 10s and 100s of different subsets from both the data set and the feature set, and trains them. With this method, hundreds of decision trees are created and each decision tree makes individual predictions. At the end of the day, if our problem is regression, if our problem is classifying the average of the estimates of the decision trees, we choose the most votes among the predictions.

6.3 Support Vector Machine (Linear)

Suppose, in classification with support vector machines, samples belonging to two classes are linearly distributed. In this case, it is aimed to distinguish these two classes with the help of a decision function obtained using training data. It is called the correct decision line that divides the data set into two. Although it is possible to draw infinite decision lines, the important thing is to determine the optimal decision line. In order for the decision line to be resistant to the newly added data, the border line must be at the closest distance to the border lines of the two classes. The points closest to this border line are called support points. Class labels in the form of (-1, + 1) are generally used in classification with support vector machines.

6.4 K-Nearest Neighbour

The K-NN (K-Nearest Neighbor) algorithm is one of the simplest and most used classification algorithm. K-NN is a non-parametric (non-parametric), lazy (lazy) learning algorithm. If we try to understand the concept of lazy, unlike eager learning, lazy learning does not have a training stage. It does not learn the training data but instead "memorizes" the training data set. When we want to make a guess, it looks for the closest neighbors in the entire data set. In the operation of the algorithm, a K value is determined. The meaning of this K value is the number of elements to look at. When a value comes, the distance between the value is calculated by taking the nearest K number of elements. The Euclidean function is generally used in the distance calculation. Manhattan, Minkowski and Hamming functions can also be used as an alternative to the Euclidean function. After the distance is calculated, it is sorted and the incoming value is assigned to the appropriate class.

6.5 CART

Classification and Decision tree (CART) learning is one of the predictive modeling approaches used in statistics, data mining and machine learning. Uses a decision tree to navigate from observations about an item to conclusions about the item's target value.

6.6 Rocchio

The Rocchio algorithm is based on a conformity feedback method found in information access systems originating from the SMART Information Retrieval System developed between 1960 and 1964. Like many other retrieval systems, the Rocchio feedback approach has been developed using the Vector Space Model.

6.7 LogisticRegression

Logistic Regression is a regression method for classification. It is used to classify categorical or numerical data. It is widely used in linear classification problems. For this reason, it is very similar to Linear Regression.

Logistic Regression is often known as binary classifications, and is only true / false, positive / negative, etc. used in binary classifications. But of course, it can turn every multi class problem into a binary classification when desired. We also used this feature of logistic regression in our multi class classification.

7 Experiment

While applying experiments, we cross validated with 10 folds.

First, the results of how each method works on different parameters will be shown. Then, a comparison of all methods will be made with their highest values.

7.1 Naive Bayes

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.8903
WITH STOPWORDS	TFIDF	500	ZembDS	0.9139
WITH STOPWORDS	TFIDF	500	F5DS	0.9144
WITH STOPWORDS	TFIDF	500	F7DS	0.9158
WITH STOPWORDS	TFIDF	1000	OrgDS	0.9089
WITH STOPWORDS	TFIDF	1000	ZembDS	0.9264
WITH STOPWORDS	TFIDF	1000	F5DS	0.9225
WITH STOPWORDS	TFIDF	1000	F7DS	0.9233
WITH STOPWORDS	TFIDF	2000	OrgDS	0.9208
WITH STOPWORDS	TFIDF	2000	ZembDS	0.9300
WITH STOPWORDS	TFIDF	2000	F5DS	0.9303
WITH STOPWORDS	TFIDF	2000	F7DS	0.9297
WITH STOPWORDS	TFIDF	5000	OrgDS	0.9261
WITH STOPWORDS	TFIDF	5000	ZembDS	0.9339
WITH STOPWORDS	TFIDF	5000	F5DS	0.9319
WITH STOPWORDS	TFIDF	5000	F7DS	0.9322
WITH STOPWORDS	TFIDF	8000 (all)	OrgDS	0.9256
WITH STOPWORDS	TFIDF	8000 (all)	ZembDS	0.9342
WITH STOPWORDS	TFIDF	8000 (all)	F5DS	0.9306
WITH STOPWORDS	TFIDF	8000 (all)	F7DS	0.9328
WITH STOPWORDS	BOW	500	OrgDS	0.8914
WITH STOPWORDS	BOW	500	ZembDS	0.9164
WITH STOPWORDS	BOW	500	F5DS	0.9136
WITH STOPWORDS	BOW	500	F7DS	0.9150
WITH STOPWORDS	BOW	1000	OrgDS	0.9083
WITH STOPWORDS	BOW	1000	ZembDS	0.9267
WITH STOPWORDS	BOW	1000	F5DS	0.9211
WITH STOPWORDS	BOW	1000	F7DS	0.9194
WITH STOPWORDS	BOW	2000	OrgDS	0.9178
WITH STOPWORDS	BOW	2000	ZembDS	0.9294
WITH STOPWORDS	BOW	2000	F5DS	0.9267
WITH STOPWORDS	BOW	2000	F7DS	0.9283
WITH STOPWORDS	BOW	5000	OrgDS	0.9289
WITH STOPWORDS	BOW	5000	ZembDS	0.9386
WITH STOPWORDS	BOW	5000	F5DS	0.9322
WITH STOPWORDS	BOW	5000	F7DS	0.9292
WITH STOPWORDS	BOW	8000 (all)	OrgDS	0.9297
WITH STOPWORDS	BOW	8000 (all)	ZembDS	0.9378
WITH STOPWORDS	BOW	8000 (all)	F5DS	0.9322
WITH STOPWORDS	BOW	8000 (all)	F7DS	0.9314
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7511
WITHOUT STOPWORDS	TFIDF	500	ZembDS	0.8783
WITHOUT STOPWORDS	TFIDF	500	F5DS	0.8697
WITHOUT STOPWORDS	TFIDF	500	F7DS	0.8581
WITHOUT STOPWORDS	TFIDF	1000	OrgDS	0.8106
WITHOUT STOPWORDS	TFIDF	1000	ZembDS	0.8997
WITHOUT STOPWORDS	TFIDF	1000	F5DS	0.8958
WITHOUT STOPWORDS	TFIDF	1000	F7DS	0.8850
WITHOUT STOPWORDS	TFIDF	2000	OrgDS	0.8608
WITHOUT STOPWORDS	TFIDF	2000	ZembDS	0.9117
WITHOUT STOPWORDS	TFIDF	2000	F5DS	0.9142
WITHOUT STOPWORDS	TFIDF	2000	F7DS	0.9047
WITHOUT STOPWORDS	TFIDF	5000	OrgDS	0.8942
WITHOUT STOPWORDS	TFIDF	5000	ZembDS	0.9267
WITHOUT STOPWORDS	TFIDF	5000	F5DS	0.9281
WITHOUT STOPWORDS	TFIDF	5000	F7DS	0.9189
WITHOUT STOPWORDS	TFIDF	8000 (all)	OrgDS	0.9042
WITHOUT STOPWORDS	TFIDF	8000 (all)	ZembDS	0.9267
WITHOUT STOPWORDS	TFIDF	8000 (all)	F5DS	0.9264
WITHOUT STOPWORDS	TFIDF	8000 (all)	F7DS	0.9272
WITHOUT STOPWORDS	BOW	500	OrgDS	0.7222
WITHOUT STOPWORDS	BOW	500	ZembDS	0.8711
WITHOUT STOPWORDS	BOW	500	F5DS	0.8636
WITHOUT STOPWORDS	BOW	500	F7DS	0.8439
WITHOUT STOPWORDS	BOW	1000	OrgDS	0.8036
WITHOUT STOPWORDS	BOW	1000	ZembDS	0.8992
WITHOUT STOPWORDS	BOW	1000	F5DS	0.8922
WITHOUT STOPWORDS	BOW	1000	F7DS	0.8803
WITHOUT STOPWORDS	BOW	2000	OrgDS	0.8544
WITHOUT STOPWORDS	BOW	2000	ZembDS	0.9119
WITHOUT STOPWORDS	BOW	2000	F5DS	0.9081
WITHOUT STOPWORDS	BOW	2000	F7DS	0.8997
WITHOUT STOPWORDS	BOW	5000	OrgDS	0.9006
WITHOUT STOPWORDS	BOW	5000	ZembDS	0.9256
WITHOUT STOPWORDS	BOW	5000	F5DS	0.9194
WITHOUT STOPWORDS	BOW	5000	F7DS	0.9242
WITHOUT STOPWORDS	BOW	8000 (all)	OrgDS	0.9061
WITHOUT STOPWORDS	BOW	8000 (all)	ZembDS	0.9261
WITHOUT STOPWORDS	BOW	8000 (all)	F5DS	0.9228
WITHOUT STOPWORDS	BOW	8000 (all)	F7DS	0.9264

7.2 Random Forest

Stopwords	Representation	Features	Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.8783
WITH STOPWORDS	TFIDF	500	ZembDS	0.9033
WITH STOPWORDS	TFIDF	500	F5DS	0.9083
WITH STOPWORDS	TFIDF	500	F7DS	0.8950
WITH STOPWORDS	TFIDF	1000	OrgDS	0.8825
WITH STOPWORDS	TFIDF	1000	ZembDS	0.9089
WITH STOPWORDS	TFIDF	1000	F5DS	0.9114
WITH STOPWORDS	TFIDF	1000	F7DS	0.9000
WITH STOPWORDS	TFIDF	2000	OrgDS	0.8875
WITH STOPWORDS	TFIDF	2000	ZembDS	0.9108
WITH STOPWORDS	TFIDF	2000	F5DS	0.9114
WITH STOPWORDS	TFIDF	2000	F7DS	0.9017
WITH STOPWORDS	TFIDF	5000	OrgDS	0.8958
WITH STOPWORDS	TFIDF	5000	ZembDS	0.9150
WITH STOPWORDS	TFIDF	5000	F5DS	0.9164
WITH STOPWORDS	TFIDF	5000	F7DS	0.9081
WITH STOPWORDS	TFIDF	8000 (all)	OrgDS	0.8903
WITH STOPWORDS	TFIDF	8000 (all)	ZembDS	0.9133
WITH STOPWORDS	TFIDF	8000 (all)	F5DS	0.9122
WITH STOPWORDS	TFIDF	8000 (all)	F7DS	0.9108
WITH STOPWORDS	BOW	500	OrgDS	0.8658
WITH STOPWORDS	BOW	500	ZembDS	0.9025
WITH STOPWORDS	BOW	500	F5DS	0.9042
WITH STOPWORDS	BOW	500	F7DS	0.8944
WITH STOPWORDS	BOW	1000	OrgDS	0.8769
WITH STOPWORDS	BOW	1000	ZembDS	0.9075
WITH STOPWORDS	BOW	1000	F5DS	0.9047
WITH STOPWORDS	BOW	1000	F7DS	0.8994
WITH STOPWORDS	BOW	2000	OrgDS	0.8856
WITH STOPWORDS	BOW	2000	ZembDS	0.9083
WITH STOPWORDS	BOW	2000	F5DS	0.9100
WITH STOPWORDS	BOW	2000	F7DS	0.9053
WITH STOPWORDS	BOW	5000	OrgDS	0.8944
WITH STOPWORDS	BOW	5000	ZembDS	0.9114
WITH STOPWORDS	BOW	5000	F5DS	0.9131
WITH STOPWORDS	BOW	5000	F7DS	0.9086
WITH STOPWORDS	BOW	8000 (all)	OrgDS	0.8950
WITH STOPWORDS	BOW	8000 (all)	ZembDS	0.9122
WITH STOPWORDS	BOW	8000 (all)	F5DS	0.9136
WITH STOPWORDS	BOW	8000 (all)	F7DS	0.9078
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7411
WITHOUT STOPWORDS	TFIDF	500	ZembDS	0.8569
WITHOUT STOPWORDS	TFIDF	500	F5DS	0.8372
WITHOUT STOPWORDS	TFIDF	500	F7DS	0.8344
WITHOUT STOPWORDS	TFIDF	1000	OrgDS	0.7656
WITHOUT STOPWORDS	TFIDF	1000	ZembDS	0.8669
WITHOUT STOPWORDS	TFIDF	1000	F5DS	0.8533
WITHOUT STOPWORDS	TFIDF	1000	F7DS	0.8517
WITHOUT STOPWORDS	TFIDF	2000	OrgDS	0.7839
WITHOUT STOPWORDS	TFIDF	2000	ZembDS	0.8678
WITHOUT STOPWORDS	TFIDF	2000	F5DS	0.8581
WITHOUT STOPWORDS	TFIDF	2000	F7DS	0.8581
WITHOUT STOPWORDS	TFIDF	5000	OrgDS	0.8081
WITHOUT STOPWORDS	TFIDF	5000	ZembDS	0.8781
WITHOUT STOPWORDS	TFIDF	5000	F5DS	0.8686
WITHOUT STOPWORDS	TFIDF	5000	F7DS	0.8656
WITHOUT STOPWORDS	TFIDF	8000 (all)	OrgDS	0.8078
WITHOUT STOPWORDS	TFIDF	8000 (all)	ZembDS	0.8764
WITHOUT STOPWORDS	TFIDF	8000 (all)	F5DS	0.8775
WITHOUT STOPWORDS	TFIDF	8000 (all)	F7DS	0.8650
WITHOUT STOPWORDS	BOW	500	OrgDS	0.7219
WITHOUT STOPWORDS	BOW	500	ZembDS	0.8444
WITHOUT STOPWORDS	BOW	500	F5DS	0.8294
WITHOUT STOPWORDS	BOW	500	F7DS	0.8253
WITHOUT STOPWORDS	BOW	1000	OrgDS	0.7647
WITHOUT STOPWORDS	BOW	1000	ZembDS	0.8617
WITHOUT STOPWORDS	BOW	1000	F5DS	0.8494
WITHOUT STOPWORDS	BOW	1000	F7DS	0.8458
WITHOUT STOPWORDS	BOW	2000	OrgDS	0.7881
WITHOUT STOPWORDS	BOW	2000	ZembDS	0.8714
WITHOUT STOPWORDS	BOW	2000	F5DS	0.8606
WITHOUT STOPWORDS	BOW	2000	F7DS	0.8597
WITHOUT STOPWORDS	BOW	5000	OrgDS	0.8128
WITHOUT STOPWORDS	BOW	5000	ZembDS	0.8772
WITHOUT STOPWORDS	BOW	5000	F5DS	0.8731
WITHOUT STOPWORDS	BOW	5000	F7DS	0.8722
WITHOUT STOPWORDS	BOW	8000 (all)	OrgDS	0.8083
WITHOUT STOPWORDS	BOW	8000 (all)	ZembDS	0.8767
WITHOUT STOPWORDS	BOW	8000 (all)	F5DS	0.8769
WITHOUT STOPWORDS	BOW	8000 (all)	F7DS	0.8725

7.3 SVM

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.8969
			ZembDS	0.9236
			F5DS	0.9222
			F7DS	0.9211
		1000	OrgDS	0.9211
			ZembDS	0.9361
			F5DS	0.9361
			F7DS	0.9356
		2000	OrgDS	0.9300
			ZembDS	0.9442
			F5DS	0.9433
			F7DS	0.9386
		5000	OrgDS	0.9356
			ZembDS	0.9492
			F5DS	0.9475
			F7DS	0.9458
		8000 (all)	OrgDS	0.9389
			ZembDS	0.9508
			F5DS	0.9506
			F7DS	0.9483
	BOW	500	OrgDS	0.8731
			ZembDS	0.8844
			F5DS	0.8869
			F7DS	0.8833
		1000	OrgDS	0.8772
			ZembDS	0.9011
			F5DS	0.8975
			F7DS	0.8925
		2000	OrgDS	0.8806
			ZembDS	0.9094
			F5DS	0.9081
			F7DS	0.8994
		5000	OrgDS	0.8942
			ZembDS	0.9192
			F5DS	0.9122
			F7DS	0.9025
		8000 (all)	OrgDS	0.9050
			ZembDS	0.9208
			F5DS	0.9186
			F7DS	0.9156
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7881
			ZembDS	0.8858
			F5DS	0.8772
			F7DS	0.8611
		1000	OrgDS	0.8306
			ZembDS	0.9022
			F5DS	0.8972
			F7DS	0.8903
		2000	OrgDS	0.8725
			ZembDS	0.9169
			F5DS	0.9147
			F7DS	0.9042
		5000	OrgDS	0.8944
			ZembDS	0.9267
			F5DS	0.9217
			F7DS	0.9206
		8000 (all)	OrgDS	0.8961
			ZembDS	0.9278
			F5DS	0.9256
			F7DS	0.9231
	BOW	500	OrgDS	0.7356
			ZembDS	0.8508
			F5DS	0.8311
			F7DS	0.8344
		1000	OrgDS	0.7886
			ZembDS	0.8592
			F5DS	0.8556
			F7DS	0.8475
		2000	OrgDS	0.8147
			ZembDS	0.8692
			F5DS	0.8603
			F7DS	0.8608
		5000	OrgDS	0.8394
			ZembDS	0.8778
			F5DS	0.8736
			F7DS	0.8778
		8000 (all)	OrgDS	0.8444
			ZembDS	0.8814
			F5DS	0.8817
			F7DS	0.8900

7.4 KNN

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.7861
			ZembDS	0.8367
			F5DS	0.8261
			F7DS	0.8189
		1000	OrgDS	0.7394
			ZembDS	0.7744
			F5DS	0.7864
			F7DS	0.7567
		2000	OrgDS	0.6092
			ZembDS	0.6886
			F5DS	0.6783
			F7DS	0.6442
		5000	OrgDS	0.7356
			ZembDS	0.8569
			F5DS	0.8244
			F7DS	0.7997
		8000 (all)	OrgDS	0.8767
			ZembDS	0.9003
			F5DS	0.9017
			F7DS	0.9006
	BOW	500	OrgDS	0.6311
			ZembDS	0.7447
			F5DS	0.7594
			F7DS	0.7100
		1000	OrgDS	0.5994
			ZembDS	0.7242
			F5DS	0.7319
			F7DS	0.6808
		2000	OrgDS	0.5744
			ZembDS	0.6914
			F5DS	0.6978
			F7DS	0.6431
		5000	OrgDS	0.5200
			ZembDS	0.6392
			F5DS	0.6244
			F7DS	0.5581
		8000 (all)	OrgDS	0.4853
			ZembDS	0.6222
			F5DS	0.5858
			F7DS	0.5386
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7383
			ZembDS	0.7919
			F5DS	0.7783
			F7DS	0.7906
		1000	OrgDS	0.7572
			ZembDS	0.7669
			F5DS	0.7286
			F7DS	0.7739
		2000	OrgDS	0.7550
			ZembDS	0.5636
			F5DS	0.5169
			F7DS	0.5958
		5000	OrgDS	0.3461
			ZembDS	0.6444
			F5DS	0.6528
			F7DS	0.5303
		8000 (all)	OrgDS	0.4969
			ZembDS	0.8639
			F5DS	0.8508
			F7DS	0.8556
	BOW	500	OrgDS	0.6506
			ZembDS	0.7269
			F5DS	0.7122
			F7DS	0.7261
		1000	OrgDS	0.6964
			ZembDS	0.7267
			F5DS	0.6989
			F7DS	0.7356
		2000	OrgDS	0.7297
			ZembDS	0.6444
			F5DS	0.6717
			F7DS	0.6950
		5000	OrgDS	0.6056
			ZembDS	0.5167
			F5DS	0.5064
			F7DS	0.5567
		8000 (all)	OrgDS	0.4997
			ZembDS	0.4517
			F5DS	0.4522
			F7DS	0.4297

7.5 CART

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.7850
			ZembDS	0.8072
			F5DS	0.8061
			F7DS	0.7944
		1000	OrgDS	0.7592
			ZembDS	0.7897
			F5DS	0.8044
			F7DS	0.7944
		2000	OrgDS	0.7472
			ZembDS	0.7894
			F5DS	0.7936
			F7DS	0.7869
		5000	OrgDS	0.7244
			ZembDS	0.7806
			F5DS	0.7856
			F7DS	0.7747
		8000 (all)	OrgDS	0.7339
			ZembDS	0.7842
			F5DS	0.7867
			F7DS	0.7753
	BOW	500	OrgDS	0.7503
			ZembDS	0.7858
			F5DS	0.7769
			F7DS	0.7925
		1000	OrgDS	0.7503
			ZembDS	0.7808
			F5DS	0.7786
			F7DS	0.7853
		2000	OrgDS	0.7592
			ZembDS	0.7794
			F5DS	0.7753
			F7DS	0.7850
		5000	OrgDS	0.7544
			ZembDS	0.7803
			F5DS	0.7761
			F7DS	0.7769
		8000 (all)	OrgDS	0.7500
			ZembDS	0.7831
			F5DS	0.7817
			F7DS	0.7811
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7114
			ZembDS	0.7653
			F5DS	0.7425
			F7DS	0.7558
		1000	OrgDS	0.7250
			ZembDS	0.7669
			F5DS	0.7436
			F7DS	0.7567
		2000	OrgDS	0.7356
			ZembDS	0.7669
			F5DS	0.7442
			F7DS	0.7572
		5000	OrgDS	0.7225
			ZembDS	0.7536
			F5DS	0.7347
			F7DS	0.7606
		8000 (all)	OrgDS	0.7106
			ZembDS	0.7539
			F5DS	0.7256
			F7DS	0.7478
	BOW	500	OrgDS	0.6778
			ZembDS	0.7464
			F5DS	0.7439
			F7DS	0.7500
		1000	OrgDS	0.7136
			ZembDS	0.7533
			F5DS	0.7489
			F7DS	0.7617
		2000	OrgDS	0.7303
			ZembDS	0.7544
			F5DS	0.7417
			F7DS	0.7661
		5000	OrgDS	0.7350
			ZembDS	0.7583
			F5DS	0.7386
			F7DS	0.7594
		8000 (all)	OrgDS	0.7178
			ZembDS	0.7525
			F5DS	0.7397
			F7DS	0.7589

7.6 Rocchio

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.8342
			ZembDS	0.8872
			F5DS	0.8761
			F7DS	0.8700
		1000	OrgDS	0.8608
			ZembDS	0.9019
			F5DS	0.8944
			F7DS	0.8892
		2000	OrgDS	0.8781
			ZembDS	0.9053
			F5DS	0.9025
			F7DS	0.8981
		5000	OrgDS	0.8931
			ZembDS	0.9106
			F5DS	0.9081
			F7DS	0.9092
		8000 (all)	OrgDS	0.8958
			ZembDS	0.9106
			F5DS	0.9106
			F7DS	0.9111
	BOW	500	OrgDS	0.4389
			ZembDS	0.5375
			F5DS	0.6003
			F7DS	0.5264
		1000	OrgDS	0.4467
			ZembDS	0.5406
			F5DS	0.6039
			F7DS	0.5275
		2000	OrgDS	0.4578
			ZembDS	0.5450
			F5DS	0.6053
			F7DS	0.5358
		5000	OrgDS	0.4661
			ZembDS	0.5472
			F5DS	0.6083
			F7DS	0.5439
		8000 (all)	OrgDS	0.4667
			ZembDS	0.5489
			F5DS	0.6086
			F7DS	0.5450
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.6542
			ZembDS	0.8297
			F5DS	0.8044
			F7DS	0.7808
		1000	OrgDS	0.7203
			ZembDS	0.8581
			F5DS	0.8364
			F7DS	0.8158
		2000	OrgDS	0.7781
			ZembDS	0.8789
			F5DS	0.8628
			F7DS	0.8422
		5000	OrgDS	0.8278
			ZembDS	0.8956
			F5DS	0.8814
			F7DS	0.8664
		8000 (all)	OrgDS	0.8333
			ZembDS	0.9006
			F5DS	0.8822
			F7DS	0.8742
	BOW	500	OrgDS	0.5314
			ZembDS	0.7047
			F5DS	0.6439
			F7DS	0.6181
		1000	OrgDS	0.5725
			ZembDS	0.7339
			F5DS	0.6608
			F7DS	0.6336
		2000	OrgDS	0.6086
			ZembDS	0.7583
			F5DS	0.6739
			F7DS	0.6467
		5000	OrgDS	0.6417
			ZembDS	0.7728
			F5DS	0.6853
			F7DS	0.6619
		8000 (all)	OrgDS	0.6497
			ZembDS	0.7778
			F5DS	0.6883
			F7DS	0.6669

7.7 Logistic Regression

			Dataset	Score
WITH STOPWORDS	TFIDF	500	OrgDS	0.8861
			ZembDS	0.9158
			F5DS	0.9142
			F7DS	0.9086
		1000	OrgDS	0.9081
			ZembDS	0.9297
			F5DS	0.9253
			F7DS	0.9231
		2000	OrgDS	0.9219
			ZembDS	0.9367
			F5DS	0.9342
			F7DS	0.9328
		5000	OrgDS	0.9278
			ZembDS	0.9428
			F5DS	0.9392
			F7DS	0.9367
		8000 (all)	OrgDS	0.9300
			ZembDS	0.9433
			F5DS	0.9417
			F7DS	0.9389
	BOW	500	OrgDS	0.8839
			ZembDS	0.9011
			F5DS	0.8994
			F7DS	0.8964
		1000	OrgDS	0.8950
			ZembDS	0.9147
			F5DS	0.9119
			F7DS	0.9097
		2000	OrgDS	0.9003
			ZembDS	0.9189
			F5DS	0.9203
			F7DS	0.9178
		5000	OrgDS	0.9094
			ZembDS	0.9247
			F5DS	0.9244
			F7DS	0.9225
		8000 (all)	OrgDS	0.9111
			ZembDS	0.9275
			F5DS	0.9269
			F7DS	0.9247
WITHOUT STOPWORDS	TFIDF	500	OrgDS	0.7786
			ZembDS	0.8750
			F5DS	0.8647
			F7DS	0.8511
		1000	OrgDS	0.8186
			ZembDS	0.8967
			F5DS	0.8892
			F7DS	0.8797
		2000	OrgDS	0.8636
			ZembDS	0.9128
			F5DS	0.9086
			F7DS	0.8997
		5000	OrgDS	0.8933
			ZembDS	0.9261
			F5DS	0.9208
			F7DS	0.9219
		8000 (all)	OrgDS	0.9006
			ZembDS	0.9297
			F5DS	0.9239
			F7DS	0.9242
	BOW	500	OrgDS	0.7361
			ZembDS	0.8569
			F5DS	0.8461
			F7DS	0.8322
		1000	OrgDS	0.8039
			ZembDS	0.8819
			F5DS	0.8731
			F7DS	0.8561
		2000	OrgDS	0.8431
			ZembDS	0.8906
			F5DS	0.8858
			F7DS	0.8847
		5000	OrgDS	0.8706
			ZembDS	0.8981
			F5DS	0.8942
			F7DS	0.9014
		8000 (all)	OrgDS	0.8725
			ZembDS	0.9003
			F5DS	0.8972
			F7DS	0.9042

8 Conclusion

First of all, based on the results of the above deep and extensive experiments, we tried to determine what is the best parameter for a method, and then to determine the common point between these parameters.

All of them featured a certain pattern. When we examine the tables, the datasets where the removal of stop-words are made, certainly performed worse. When we compare the lines, stop-word, i.e. conjunction, etc. texts from which words were removed always showed less successful results. We have seen in all experiments without exception that removing the stop-words lowers the scores.

A second pattern is that the TFIDF vector generally performs better than the BOW vector. TFIDF lines gave more successful results than BOW lines in the scores of the parameter experiments we performed for all methods. This does not apply to the Naive Bayes alone. Naive Bayes provided the best score for all parameters with the BOW vector. But still, we can consider the success of TFIDF.

Among the best 7 scores, 4 are the result obtained on Zemb-DS. Two of them were obtained on F5-DS and the last one on F7-DS.

We can see from the general pattern analysis we made about the results that the dataset processed by stemming with the spring wire has generally yielded more successful results.

The best results, together with these common patterns we have caught with; using the TFIDF in the attribute vector, the preeprocessed data set is stuck in the resulting range, with the stop-words not extracted. So now, in order to gain a more distant view of the picture, we can draw the following table, which only takes this range into our perspective.

	Feature: 500				Feature: 5000				Feature: 8000 (All)			
	OrgDS	ZembDS	F5DS	F7DS	OrgDS	ZembDS	F5DS	F7DS	OrgDS	ZembDS	F5DS	F7DS
NB	0.8903	0.9139	0.9144	0.9158	0.9261	0.9339	0.9319	0.9322	0.9256	0.9342	0.9306	0.9328
RF	0.8783	0.9033	0.9083	0.8950	0.8958	0.9150	0.9164	0.9081	0.8903	0.9133	0.9122	0.9108
SVM	0.8969	**0.9236**	0.9222	0.9211	0.9356	**0.9492**	0.9475	0.9458	0.9389	0.9508	**0.9506**	0.9483
KNN	0.7861	0.8367	0.8261	0.8189	0.7356	0.8569	0.8244	0.7997	0.8767	0.9003	0.9017	0.9006
CART	0.7850	0.8072	0.8061	0.7944	0.7244	0.7806	0.7856	0.7747	0.7339	0.7842	0.7867	0.7753
Rocchio	0.8342	0.8872	0.8761	0.8700	0.8931	0.9106	0.9081	0.9092	0.8958	0.9106	0.9106	0.9111
LR	0.8861	0.9158	0.9142	0.9086	0.9278	0.9428	0.9392	0.9367	0.9300	0.9433	0.9417	0.9389

In the table below, the best scores obtained in certain data sets in the reference study in all methods and the best scores obtained in the specific dataset in our study are compared.

	[1]				Ours			
	OrgDS	ZembDS	F5DS	F7DS	OrgDS	ZembDS	F5DS	F7DS
NB	0.8294	0.8719	0.8222	0.8403	0.9256	**0.9342**	0.9319	0.9328
RF	0.8887	0.9103	0.8828	0.8859	0.8903	**0.9150**	0.9122	0.9108
SVM	0.8603	0.8497	0.8239	0.8356	0.9389	0.9492	**0.9506**	0.9483
KNN	0.7311	0.7497	0.6944	0.7256	0.8767	0.9003	**0.9017**	0.9006
CART	0.7897	0.7939	0.7736	0.7597	0.7850	0.8072	**0.8061**	0.7944

Better results were obtained in our study in all methods and in all datasets.

9 References

1 Kilinç, Deniz & Ozcift, Akin & Bozyiğit, Fatma & Yildirim, Pelin & Yucalar, Fatih & Borandağ, Emin. (2017). TTC-3600: A new benchmark dataset for Turkish text categorization. Journal of Information Science. 43. 174-185.

2 A. Fallis, “Text Categorisation: A Survey,” J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 1999.

3 M. U. SALUR and İ. AYDIN, ”The Impact of Preprocessing on Classification Performance in Convolutional Neural Networks for Turkish Text,” 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-4.

4 İ. Aydın, M. U. Salur and F. Başkaya “Duygu Analizi için Çoklu Popülasyon Tabanlı Parçacık Sürü Optimizasyonu,” Türkiye Bilişim Vakfı Bilgi. Bilim. ve Mühendisliği Derg., vol. 11, no. 1, pp. 52–64, 2018.

5 Can F, Kocberber S, Balcik E, Kaynak C, Ocalan HC and Vursavas OM. Information retrieval on Turkish texts. Journal of the American Society for Information Science and Technology 2008; 59(3): 407–421.

6 Akin AA and Akin MD. Zemberek, an open source NLP framework for Turkic Languages. Structure 2007; 10: 1–5.