Volume 1 Issue 2 2023
Pages: 122-138
ISSN: 2980-3152

CURRENT TRENDS IN COMPUTING

http://ctc.karabuk.edu.tr
RESEARCH ARTICLE

TEXT CLASSIFICATION WITH FREQUENCY-BASED TEXT VECTORIZATION METHODS FOR ENHANCING CALL CENTER EFFICIENCY

MUAMMER ÖZDEMİR[1], YASİN ORTAKCI[1*]

[1]*Computer Engineering Department, Karabük University, 78050, Karabük, Türkiye*

ABSTRACT: In today's business world, many transactions take place over the phone or online. Call centers play a significant role in dealing with different situations and solving problems that come with the large volume of global business. As an interface between companies/institutions and customers, call centers aim to eliminate problems, correct mistakes, resolve conflicts, and increase customer satisfaction. The traditional approach involves customer service agents handling inquiries and complaints, but human error can hinder effective problem resolution. Intelligent assistant applications have emerged to augment the skills of customer service agents, improve performance, and maximize customer satisfaction. This study focuses on addressing the challenges faced by the Republic of Turkiye Ministry of Trade Call Center (RTMTCC), which handles over 10,000 calls per day. For this purpose, it introduces an intelligent framework that uses AI-driven methods and frequency-based text vectorization techniques to efficiently route calls to relevant departments, with the aim of increasing customer satisfaction and reducing economic losses. Using historical call texts, Bag of Words (BoW), and Term Frequency-Inverse Document Frequency (TF-IDF), the study evaluates the performance of five different classifiers: Stochastic Gradient Descent (SGD), Logistic Regression (LR), Naive Bayes (NB), Adaptive Boosting (AdaBoost), Artificial Neural Networks (ANN). The results indicate that the AdaBoost classifier generally outperforms others in both text vectorization approaches by reaching higher precision, recall and F1-score values. The study provides new approaches to automate call routing, evaluates how to classify text effectively, and shows the strengths and weaknesses of different text analysis methods.

1. INTRODUCTION

In today's business world, many transactions are carried out remotely over the telephone call or the internet [1]. The importance of call centers in handling new situations and solving problems arising from the massive volume of business in the world is increasing day by day [2]. Call centers serve as an interface department between companies&institutions and customers to eliminate problems, mistakes, conflicts, and misunderstandings and to increase customer satisfaction. Call centers receive customer requests both through telephone calls and Internet messages.

In the traditional call center approach, customer requests and complaints are handled by customer service representatives and problems are attempted to be solved with the efforts and knowledge of these customer service representatives. However, due to human errors such as high workload, stress, distraction [3], and lack of experience and knowledge [4], customer problems may not be solved or may take a long time to be solved. It has also been observed that call centre agents are at higher risk of burnout and job dissatisfaction than employees in other sectors [5].

E-mail address: yasinortakci@karabuk.edu.tr (*)
Key words and phrases. Text classification, Text vectorization, TF-IDF, BoW, Call centers, Customer service.

Today, many smart assistant applications have been developed to help customer service representatives overcome these problems. Through these smart applications, the performance of customer service agents is improved, and customer satisfaction is maximized.

In this context, the study aims to tackle issues encountered at the Republic of Turkiye Ministry of Trade Call Center (RTMTCC). On average, the RTMTCC receives over 10,000 calls each day, which are either responded to manually by customer representatives or directed to one of the 99 departments within the Ministry. Due to the complexity of the call content or an error by the customer service representative, many of these calls are often directed to the incorrect department. Consequently, this results in prolonged problem solutions and causes economic losses for the country.

To address these challenges, this study presents an intelligent framework designed to efficiently route incoming calls to relevant departments at the RTMTCC through the application of AI-driven methods and frequency-based text vectorization techniques. The primary goal of this approach is to increase customer satisfaction and mitigate economic losses by improving the speed and reliability of call resolution. In this context, we used a sample dataset consisting of historical call texts received by the Ministry's call center, along with information on the corresponding departments to which they were directed. The call texts were transformed into digital representations using Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) text representation methods. We then created different classifier models by training classification algorithms such as Stochastic Gradient Descent (SGD), Logistic Regression (LR), Naive Bayes (NB), Adaptive Boosting (AdaBoost), and Artificial Neural Networks (ANN) on our sample dataset.

In addition to creating a smart customer service system, one of the main research questions of this study is to identify the most effective combination of classification algorithms and text vectorization methods for the purpose of routing call recordings from the RTMTCC to the relevant departments. To achieve this goal, we evaluate the performance of each classification algorithm individually using both BoW and TF-IDF methods. This evaluation is carried out through 5-fold cross-validation, with a thorough examination of accuracy, precision, recall, and F1 scores. As a result, our study shows that the AdaBoost classifier outperforms other methods in both text vectorization approaches on three out of four performance metrics. In addition, the contributions of this study can be summarized as follows:

- It presents an automated call routing system within the RTMTCC, which streamlines the workflow of call center agents and efficiently routes calls to the appropriate departments.
- A comprehensive evaluation of text classification performance of various classifiers, including SGD, LR, NB, AdaBoost, and ANN, in conjunction with both BoW and TF-IDF text vectorization techniques are performed.
- The strengths and limitations of TF-IDF and BoW as text vectorization approaches are revealed. These findings contribute to a deeper understanding of text processing and routing optimisation in call center operations.

2. LITERATURE REVIEW

Call centers have become critical communication hubs for today's businesses, providing key touch points for customer interactions. Managing incoming calls is critical to improving customer satisfaction, optimising resource allocation and streamlining business operations. In this context, there has been a recent increase in research into call center performance and the assistive applications used in this area.

One of the fundamental issues in call center operations is how to classify incoming calls and route them to the correct department. Chatterjee et al. classify calls as problematic or non-problematic using Support Vector Machines (SVM) and highlight the potential of their method to address call complexity. The proposed system achieved an accuracy of 87.5% in identifying problematic calls during experiments [6]. Mishne et al. presented a system that integrates transcription of call center conversations with analysis of their content, providing knowledge-mining tools for agents and administrators. Initial experiments with the system, based on manually transcribed data, yield promising results [7]. Galanis et

al. conducted a study on classifying emotional speech during call center interactions and its implications for predicting emotions with natural language processing (NLP The study compares the performance of SVM classifiers using different radial bases function kernels such as SVM-RBF1 and SVM-RBF2 in classifying emotional utterances in call center interactions. The study shows that SVM-RBF2 achieves higher accuracy for almost all attempts, as the speaker role attribute has the highest Pearson correlation value [8]. Rashid et al. presented a study focused on increasing sales and reducing costs through intelligent use of data and human resources by improving call center performance. They used data mining techniques to analyse and implement the 3R concept (Right Data to the Right CSR at the Right Time) for sales optimisation. The implementation results demonstrate that integrating the 3R concept into outbound call centers resulted in substantial enhancements in sales volume, revenue, and overall productivity. Furthermore, it led to reduced expenses, significantly decreased the number of dialled contacts, and witnessed a marked decrease in the involvement of customer service representatives during the campaign [9].

In the area of studying text classification methods and evaluating the effectiveness of text classifiers, a variety of scholarly investigations contribute important insights to the field. Yigit et al. focused on predictive models for evaluating speech recordings in call center text mining, including customer satisfaction measurement and sentiment analysis using classification and regression techniques. Various classification algorithms, including Decision Tree (DT), SVM, K-Nearest Neighbor (KNN), LR, and Random Forest (RF), were tested to determine the most effective algorithm for classifying the sentiments of an interaction. The results showed that the SVM algorithm was the most successful with a slight difference, achieving an accuracy rate of 82% [10]. Fiok et al. introduced a text truncation technique called text guide, which is designed to condense the original text to a predetermined limit. This method improves performance over naive and semi-naive approaches while minimising computational complexity. The text guide uses the concept of feature importance, a key principle in explainable artificial intelligence, to optimise its functionality. The authors conducted a careful analysis and subsequent experiments to evaluate the effectiveness of the text guide. Their results showed that the text guide showed significant improvements, especially when the length of the original text instances significantly exceeded the model boundary [11]. In another study, Chen et al. focused on legal text classification using labelled case documents and comparing different techniques. They propose a machine learning algorithm that incorporates domain concepts as features and uses RF as classifiers. Through experiments on 30,000 case documents across 50 categories, their approach outperforms a deep learning model based on pre-trained word embeddings and neural networks. The domain concept-based RF classifier demonstrates superior performance compared to its deep learning-based counterpart in a variety of experimental settings in terms of accuracy, recall, precision, and F1 score [12].

Given the limited availability of libraries and resources for Turkish, the need for research in the field of NLP becomes obvious. Consequently, any research that deals with text and document classification in Turkish acquires a heightened significance. There have been some remarkable studies carried out in this regard. For instance, Sarı used deep learning methods, Paragraph Vector Distributed Memory Model (PV-DM), Vector-Distributed Bag of Words, and Doc2Vec to classify columns and predict authors and concluded that the PV-DM model is the most effective [13]. Koruyan and Ekeryilmaz analysed complaints on "sikayetvar.com" using machine learning for automatic categorisation and analysis and achieved 80% accuracy with LR [14]. Uslu and Akyol classified Turkish news texts using SVM, RF, and NB in their study, in which NB was more successful with a 91% F1 score than others [15]. Karakuş et al. analyzed the readability of Turkish primary school textbooks using a distributed parallel processing framework based on the MapReduce model. They demonstrated the practicality and efficiency of the results. The study demonstrated the distributed system's efficiency and feasibility in analyzing a significant number of Turkish textbooks' readability. [16]. Kuzu et al. examined how converting 2560 phone records into text format impacted content classification in a call center subject recognition study. They assessed the macro and micro F1 values of K-Means, ANN, and SVM. The results showed that SVM and ANN produced similar text classification curves [17].

The previous studies have highlighted previous efforts in Turkish text call classification. Our primary objective is to enhance the field of Turkish text call classification by utilizing call center data from the RTMTCC. To accomplish this objective, we completed a comprehensive study comparing the TF-IDF and BoW methodologies, as well as different classification algorithms.

3. DATA AND METHODS

Figure 1 shows a general overview of our text classification framework. First, we created our dataset by transcribing 100,000 phone calls received by the RTMTCC for this study. Data preprocessing removes the redundant and noisy data from the call text. Text vectorization transforms the text into a numerical representation using the BoW and TF-IDF text vectorization methods. In the last step, different classifier models assign the text representations to the relevant departments. No feature selection procedure was carried out, since some sparse terms in our dataset may have a notable impact on the classificaiton of many records. Hence, all available features were utilized in the classification task.

FIGURE 1. Overview of the proposed intelligent text classification framework.

3.1 Dataset

Our dataset is a new collection of call records from the RTMTCC between customer service representatives and customers. It includes 100,000 calls from a randomly chosen period, categorized into 99 distinct classes without any restriction on the number of characters. These call records were transcribed by call center employees, and a corpus dataset was created. Table A.1 in the Appendix section lists the distribution of calls across different departments, presenting the average word and character counts for each department. Unfortunately, the confidentiality of the dataset prevents it from being shared, as it contains sensitive data from the Republic of Turkiye Ministry of Trade. Since it contains institutional information associated with the Ministry, the names of the departments are labeled numerically from 1 to 99.

Figure 2 illustrates the average number of words and characters of the phone calls for the 25 classes with the highest number of call records. As the workflow in some departments is more complex, the phone calls for these departments take longer durations, and therefore, the average number of words and characters for these departments is higher. After data preprocessing, both TF-IDF and BoW vectorization methods transformed the dataset to a vector space, producing 11,207 unique features, each representing a different Turkish term found in the corpus.

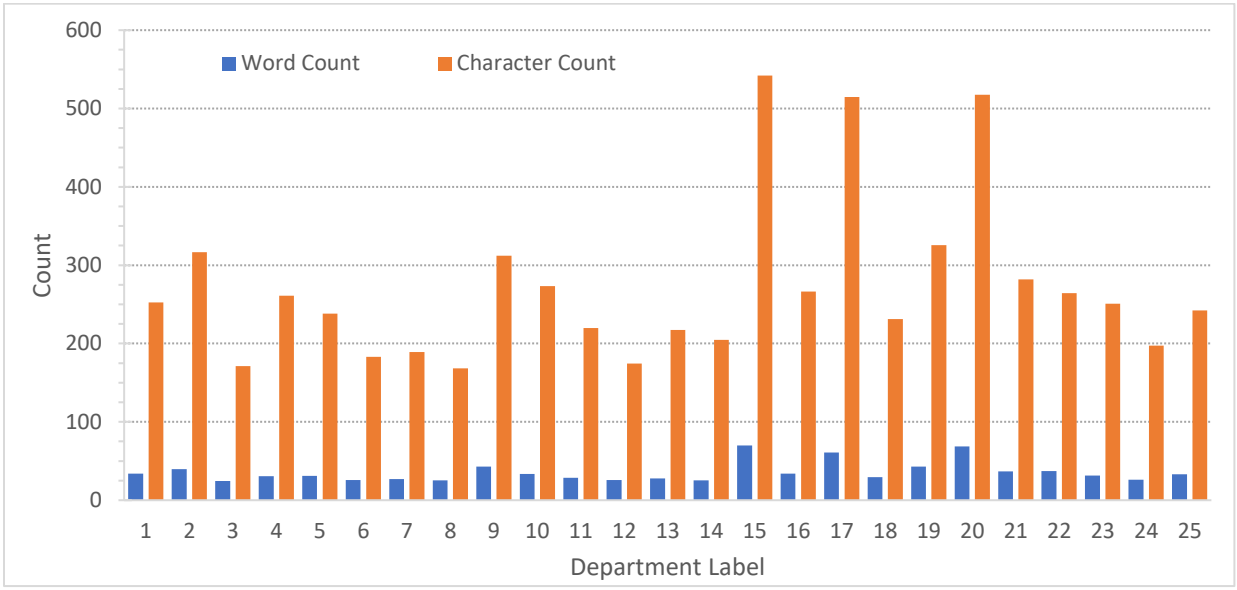

FIGURE 2. Average word and character counts of the 25 departments with the highest number of telephone calls.

3.2 Data Preprocessing

Data preprocessing plays a critical role in the construction of our classification framework, including essential tasks such as data cleaning, tokenization and lemmatization. The first phase of data cleaning involves correcting errors, addressing inconsistencies, handling missing values, and eliminating duplicates and empty entries within the call center record dataset. We then standardized all call text to lowercase, removed punctuation and identified and extracted user-specific details such as ID numbers, telephone numbers and customs declaration numbers. We also removed the call text from stop words, which are common words that have no significant meaning [18]. We excluded 53 stop words using the "Turkish" class within the Natural Language Toolkit (NLTK) library[1]. The NLTK library, a versatile open-source Python toolset, provides a range of functions for natural language processing (NLP), artificial intelligence and information retrieval[2].

The next stage in the data preprocessing flow is tokenization, which splits the text into meaningful elements such as words, phrases, or symbols called tokens. Tokenization facilitates the representation of text in a structured format, enhancing the usefulness of text analysis [19].

The final stage of data preprocessing was lemmatization, an NLP technique that transforms words by replacing or removing suffixes to obtain their basic forms or lemmas [20]. In this study, we used the 'tr' class of the Simplemma library, an open-source Python library that supports multiple languages and helps find root words [21]. By applying lemmatization, we extracted more concise information from our dataset and prepared it for further analysis and model building.

3.3 Text Vectorization

Converting text data into vectors is a way to communicate with the machines to perform any NLP tasks and solve problems mathematically [22]. For text vectorization, we used two different frequency-based methods, BoW and TF-IDF, which transform texts into numerical representations.

3.3.1 Bag of Words

BoW is a technique for extracting features from text for further NLP analysis. BoW operates by considering the occurrence and repetition of words within a text without taking into account their semantic meaning, contextual relevance, or order. Each unique word present in the call record corpus

[2] https://github.com/xiamx/node-nltk-stopwords
[3] https://www.nltk.org/

corresponds to a distinct feature. When a word is found in the call text, it is assigned a non-zero value. In addition, if a word recurs within a text, the BoW value for that word corresponds to its frequency in the text. For example, if a word is repeated four times in a text, the value of that word in the BoW vector is four [23]. Table 1 illustrates a BoW representation of a sample text.

TABLE 1. A sample representation of the BoW method for a sample text.

Text ID	**Words**							
	al	almanya	ara	kapıkule	bir	cam	liman	kooperatif
0	0	0	1	0	1	0	0	0
1	0	0	0	0	2	0	0	3
2	0	0	0	0	0	0	1	0
3	0	1	4	0	1	0	0	0
4	1	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	1
6	0	1	0	1	0	0	0	0

3.3.2 Term Frequency-Inverse Document Frequency

One of the main drawbacks of BoW is that common words become dominant in the text vectors. These frequent words, which do not have any significant semantic impact in the texts, can overwhelm other features and reduce their influence. To address this issue, TF-IDF calculates both the term frequency and inverse document frequency for text vectors. TF-IDF considers not only the number of times a word appears in a document but also its importance in the entire corpus. While term frequency (TF) calculates the frequency of a word in a document, inverse document frequency (IDF) measures the rarity of the word across all documents [24]. The assigned score for a word in a document is proportional to the product of its TF and IDF scores, ranging from 0 to 1. Words that occur frequently in a document but rarely in all documents receive a high score, indicating their significance. Conversely, words that occur frequently in all documents receive a low score, indicating their lack of relevance.

The TF is determined by dividing the number of times a term appears in a document by the number of words in that document, as shown in Eq. (1). On the other hand the IDF indicates how frequently a term appears in the collection of documents. Terms that are specific to documents have higher IDF values compared to common words. To calculate the IDF, we need to divide the total number of documents by the number of documents containing a word and then take its logarithm as formulated in Eq. (2).

To illustrate the practical calculation of a straightforward TF-IDF value within a given text, let's consider an example. The TF-IDF value is obtained by multiplying TF and IDF as shown in Eq. (3).

$$TF = \frac{number\ of\ times\ the\ term\ apears\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

$$IDF = \log\left(\frac{number\ of\ the\ documents\ in\ the\ corpus}{number\ of\ documents\ in\ the\ corpus\ contain\ the\ term + 1}\right) \quad (2)$$

$$TF - IDF = TF * IDF \quad (3)$$

For example, if the word "kapikule" occurs four times in a text of 50 words, the TF value is 0.08. Similarly, if the word "kapikule" appears in 10 texts out of 500 texts, the IDF value is 1.69. By multiplying these two values, the TF-IDF value is calculated as 0.135. Table 2 shows a sample score table of the TF-IDF method.

TABLE 2. A sample representation of the TF-IDF method for a sample text.

Text ID	Words							
	al	almanya	ara	kapıkule	bir	cam	liman	kooperatif
0	0	0	0,290	0	0,280	0	0	0
1	0	0	0	0	0,255	0	0	0,396
2	0	0	0	0	0	0	0,587	0
3	0	0,319	0,478	0	0,325	0	0	0
4	0,474	0	0	0	0	0	0	0
5	0	0	0	0	0	0,454	0	0,482
6	0	0,280	0	0,135	0	0	0	0

3.4 Machine Learning Classifiers

In this study, we perform a comprehensive performance analysis of different classification algorithms with a combination of text vectorization techniques. These classifiers are developed by using SGD, LR, NB, AdaBoost, and ANN methods.

3.4.1 Stochastic Gradient Descent (SGD)

SGD is a widely used sequential optimization algorithm for both regression and classification tasks [25]. It provides an efficient and iterative approach to minimizing a loss function and determining the optimal parameters. SGD is also known for its versatility, being suitable for both linear and non-linear classification tasks. As a variant of the gradient descent optimization algorithm, SGD differs from the traditional gradient descent method in a crucial way. Instead of computing the gradient of the loss function using the entire training data at each iteration, SGD calculates the gradient using only one training example or a small random subset of examples at a time. This unique approach introduces an element of randomness into the parameter updates, which can be beneficial to the algorithm. This stochasticity helps the algorithm escape local minima, allowing for faster convergence, especially when working with large datasets.

3.4.2 Logistic Regression (LR)

LR, a widely used statistical and machine learning method, serves as a multipurpose classification algorithm. LR excels at handling binary classification tasks involving target variables with two classes, as well as more complex multiclass classification problems involving more than two classes [26]. In binary classification scenarios, it is often referred to as "binary logistic regression". In contrast, in multiclass classification situations, it is extended and referred to as "multinomial logistic regression" or "softmax regression".

3.4.3 Naïve Bayes (NB)

NB is an effective machine learning algorithm for classification and text categorization tasks. It is based on Bayes' theorem and is particularly useful in NLP and document classification tasks [27]. The term "naive" in Naive Bayes implies an assumption of conditional independence between features. It presumes that each characteristic is autonomous of others, provided the class label. NB has the potential to work exceptionally well, particularly with textual data. NB is frequently employed in text classification, such as identifying unwanted messages, analyzing feelings, and categorizing subjects. It determines the likelihood of certain words or characteristics appearing in a document based on the document's category. Types of NB classifiers:

- Multinomial NB: Often used for text classification tasks. It models the probability distribution of word occurrences in a document.
- Bernoulli NB: Suitable for binary data where features are either present or absent.
- Gaussian NB: Assumes that features follow a Gaussian distribution. It is used for continuous data.

3.4.4 Adaptive Boosting

Adaptive Boosting, or AdaBoost for short, is a powerful ensemble learning method commonly used in machine learning for regression and classification tasks. Its primary goal is to improve the accuracy of weak learners, such as classifiers or regressors, by combining their predictions into a strong and precise model [28].

3.4.5 Artificial Neural Networks (ANN)

ANN, or artificial neural networks, is a versatile and powerful method of machine learning that forms the foundation of deep learning and many other machine learning models. These neural networks can effectively capture complex patterns and representations in data, making them essential for a diverse range of tasks in modern artificial intelligence and machine learning [29].

4. EXPERIMENTAL STUDY

4.1 Setup

The experiment was carried out on an HP ProOne 440 G6 24-inch all-in-one desktop computer, boasting an i7-10700 CPU operating at 2.90GHz, 2904 MHz, 8 cores, and 16 logical processors, complemented by 32 GB RAM. Python served as our programming language and code compilation was facilitated by Spyder. Data preprocessing was a crucial step involving the utilization of various libraries such as Pandas, Numpy, NLTK, and Simplemma.

Table 3 indicates the optimal parameters used in this study for each classifier. Meanwhile, the remaining parameters were set to their default values in Python NLTK libraries.

TABLE 3. The hyperparameter values of each classifier used in the experiment.

Method	Parameter	Value
LR	max_iter	1000
NB	alpha	1.0
AdaBoost	algorithm	SAMME
	base_estimator	ExtraTreeClassifier
	n_estimators	50
ANN	hidden_layer_sizes	30
	loss function	lbfgs
	transfer (activation) function	relu
	alpha (regularization term)	0.00001
	max_iter	1000
	ANN type	feed forward

4.2 Performance Metrics

Machine learning includes a wide variety of algorithms that have been developed with a focus on specific data characteristics [30]. In this study, we investigate the performance of algorithms for text classification and vectorization using key metrics from the confusion matrix, a widely accepted tool for evaluating the performance of classification models, as shown in Figure 3. We measured accuracy, precision, recall, and F1 score for the multiclass scenario. The confusion matrix numerically represents predicted and true values, distinguishing between true negative (TN) and true positive (TP) instances and false positive (FP) and false negative (FN) instances [31].

The formula for accuracy, precision, recall, and the f1-score calculated in the light of the confusion matrix are given in Eq. (4-7). Accuracy, precision, recall and f1-score metrics take values in the range of [0,1] where values closer to 1 indicate more favorable classification outcomes. The optimal values for these metrics can differ based on the specific application field. Regarding text classification, most studies in literature have achieved an accuracy rate surpassing 80%. Therefore, 0.8 can be considered as a notable evaluation threshold within the text classification studies.

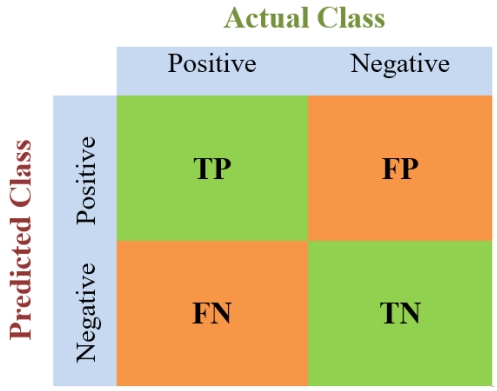

FIGURE 3. General elements of a confusion matrix in binary classification.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

To ensure unbiased evaluations, we employed cross-validation, a statistical technique. The primary aim is to facilitate the robust generalization of a machine learning model, addressing issues like overfitting and dataset-specific biases. This is achieved by evaluating the model's performance across diverse data partitions, thereby avoiding reliance on a single split. For reliability of the experimental results, we applied 5-fold cross-validation to each model, by dividing the dataset into five separate partitions. During each round, one partition functions as the test set, while the remaining four are designated for training. This comprehensive evaluation across multiple data splits provides a holistic understanding of the model's overall performance. The results are presented with aggregation of different folds.

4.3 Results and Discussion

In this study, we run each classifier ten times on the dataset, using both BoW and TF-IDF text representation methods, respectively. The accuracy with standard deviation (SD), precision, recall, and f1-score results of these experiments are presented in Table 4. Considering all the experimental results, the combination of ANN and BoW methods achieved the highest accuracy rate of 96.7% in classifying the call text to the correct department. On the other hand, in terms of precision, recall, and f1-score metrics, the TF-IDF/AdaBoost duo yielded superior precision, recall, and F1 score.

TABLE 4. Text classification results of each classifier for the text vectorization methods.

Method	Classifier	Accuracy ± SD	Precision	Recall	F1 Score
TF-IDF	SGD	0,9547 ± 0,0027	0,5278	0,4588	0,4762
	LR	0,9606 ± 0,0014	0,5627	0,4917	0,5139
	NB	0,8977 ± 0,0017	0,2902	0,1856	0,1958
	AdaBoost	0,9539 ± 0,0051	**0,642**	**0,5329**	**0,559**
	ANN	**0,9647** ± 0,0025	0,4671	0,5013	0,4721
BoW	SGD	0,9605 ± 0,0019	0,4316	0,3772	0,3921
	LR	0,9665 ± 0,0012	0,5367	0,451	**0,4762**
	NB	0,9106 ± 0,0027	0,2921	0,185	0,2004
	AdaBoost	0,954 ± 0,0026	**0,5611**	0,4394	0,4713
	ANN	**0,967** ± 0,0036	0,4724	**0,4809**	0,4731

Figure 4 illustrates the accuracy and error bar values for both vectorization methods across all classifiers. The ANN outperformed all other classifiers in accuracy parameters for both TF-IDF and BoW methods, while most classifiers exhibited closely aligned accuracy values, with the exception of NB. NB yielded the lowest classification accuracy values in both text vectorization methods, with 0.8977 for TF-IDF and 0.9106 for BoW. Additionally, when we examine the effects of TF-IDF and BoW methods on the classifiers, we realize that BoW achieves slightly higher accuracy than TF-IDF in all classifiers. Furthermore, the remarkably low SD values in each classifier indicate a high level of consistency in the classification results.

Figure 5 depicts the precision and error bar values of the experiments. Particularly, TF-IDF/AdaBoost was found to be the most favorable combination, surpassing all other pairs in this metric. The performance of NB in both text vectorization methods still remains much lower than other classifiers for precision. On the other hand, Figure 6 displays recall metric results in which the TF-IDF/AdaBoost pair produced the optimal results while NB yielded the poorest ones as in precision results. Similarly, the f1-score results given in Figure 7 reinforce the fact that the AdoBoost/TF-IDF pair outperforms other methods in text classification, with Naive Bayes (NB) emerging as the least successful approach in this context.

Although the ANN/BoW pair achieves the highest accuracy in classification, AdoBoost/TF-IDF is superior in the remaining three classification parameters. Since the distribution of 100,000 records to 99 different departments in our dataset is significantly unbalanced, the f1-score is a more dependable metric than accuracy. Therefore, it is evident that the AdoBoost/TF-IDF combination is the most effective method to classify RTMTCC call texts. In addition, classifiers using the TF-IDF approach generally outperform those that use BoW. Furthermore, while different classifiers excel in various metrics with the BoW method, AdaBoost consistently outperforms others in three out of the four metrics in the TF-IDF method. This confirms that AdaBoost/TF-IDF stands out as the superior choice, consistently delivering reliable outcomes.

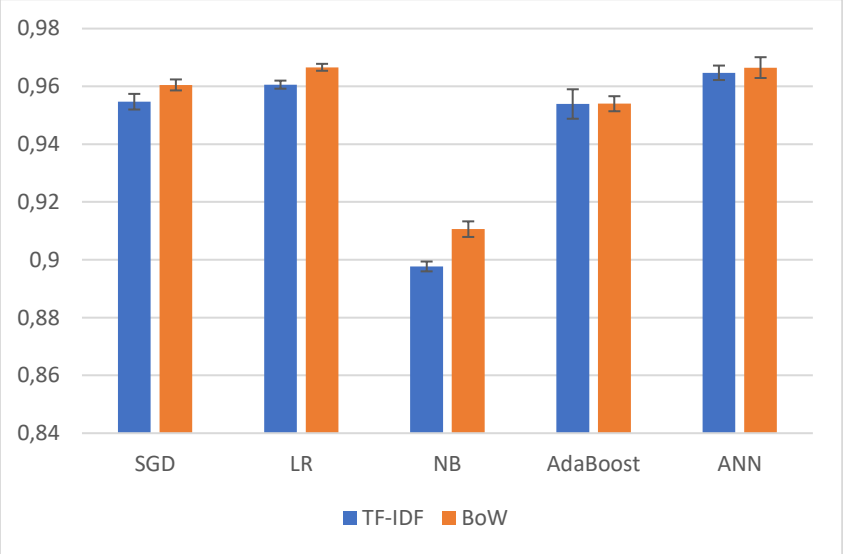

FIGURE 4. Accuracy results and error bars obtained from each classifier for BoW and TF-IDF.

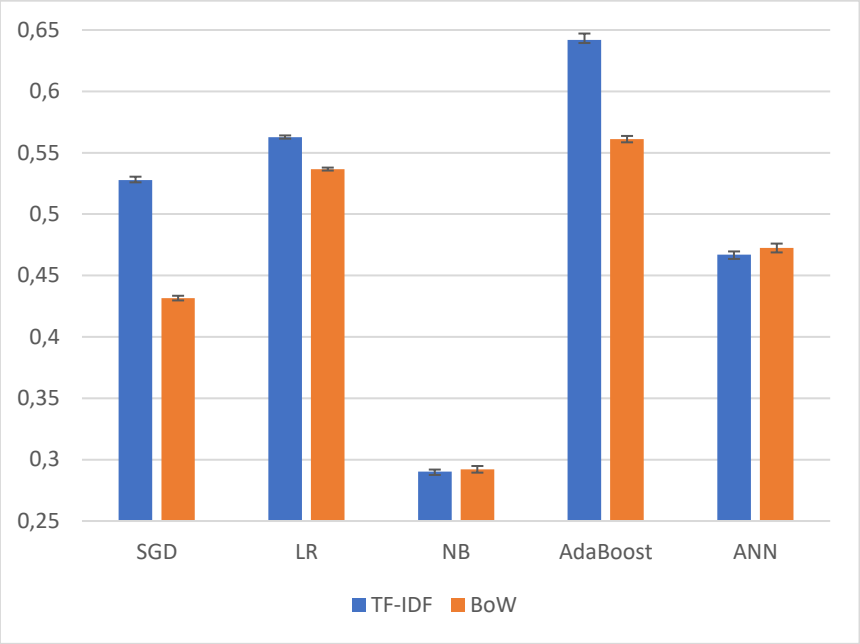

FIGURE 5. Precision results and error bars obtained from each classifier for BoW and TF-IDF.

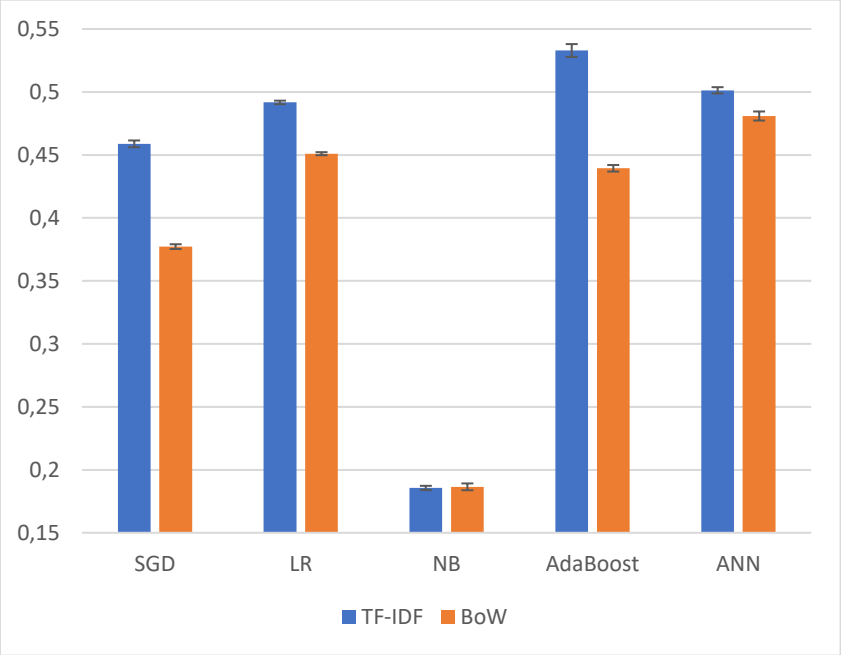

FIGURE 6. Recall results and error bars obtained from each classifier for BoW and TF-IDF.

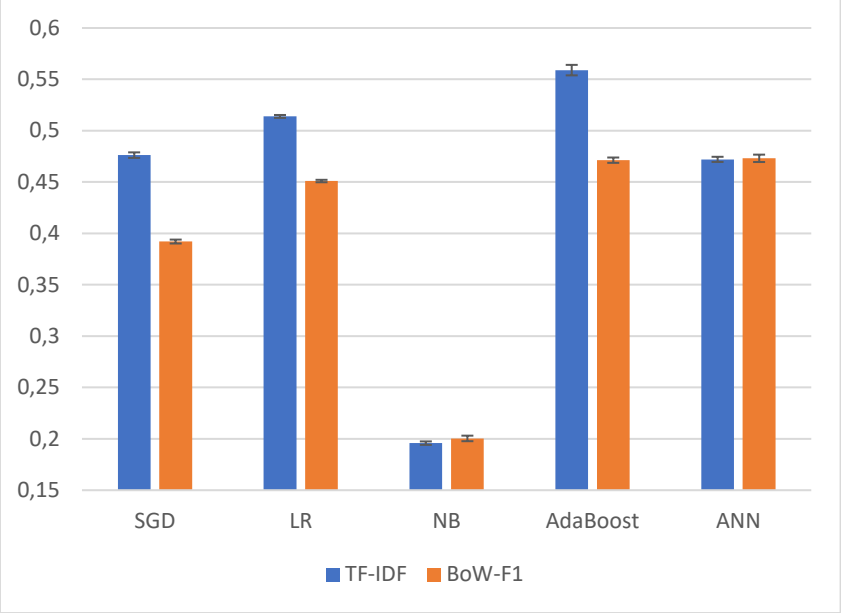

FIGURE 7. F1-Score results and error bars obtained from each classifier for BoW and TF-IDF.

As our investigation is concerning about Turkish text, we conducted a comparative analysis with existing studies that focus on the classification of Turkish text. For instance, in [14], SVM and TF-IDF demonstrated an 80% accuracy in classifying customer complaints, while [32] reported an 87% accuracy in sentiment analysis using the SVM and TF-IDF combination. Likewise, [33] highlighted a 90% accuracy in spam detection with the Gradient Boosting/TF-IDF pair, and [34] achieved a remarkable 95% accuracy in classifying Turkish news texts using SVM/FastText. In contrast, as illustrated in Figure 8, our study outperformed these benchmarks in Turkish text classification, achieving a notable 97% accuracy with the ANN/BoW combination on a large dataset containing more records than the datasets in other studies. In light of these findings, it is evident that our study constitutes a significant contribution to the field of Turkish text classification.

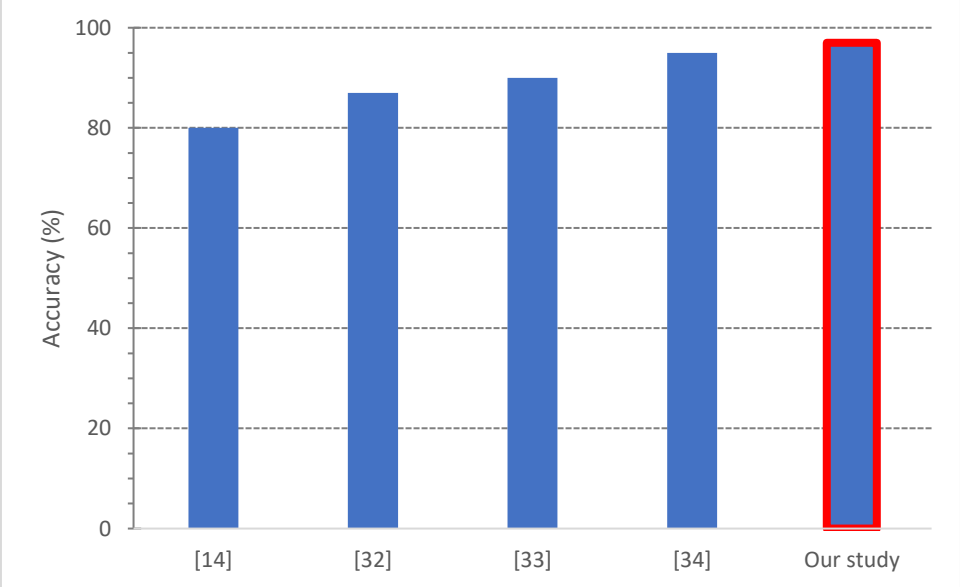

FIGURE 8. Comparison of our study with the other Turkish text classification studies in the literature for accuracy metric.

5. CONCLUSION

In this study, the efficacy of several prominent machine learning algorithms, including SGD, LR, NB, AdaBoost, and ANN, in the task of classifying 100,000 textual call records received by the RTMTCC have been tackled. Additionally, both BoW and TF-IDF text vectorization techniques to transform the call texts into numerical representations were applied. This allowed us to assess the performance of different combinations of machine learning classifier algorithms and text vectorization methods. As a result of the extensive analysis, the results have demonstrated the superiority of the AdaBoost classifier in accurately classifying call recordings, regardless of the vectorization method employed. Furthermore, the performance of other classifiers, apart from NB, is relatively close to AdaBoost. The findings also indicate that TF-IDF can be referred to as the BoW method for text classification tasks.

Throughout our study, we encountered several notable challenges. These challenges included issues related to text vectorization, the lack of efficient preprocessing tools for Turkish texts, the lack of research on Turkish text classification, and the significant computational cost. In order to overcome these obstacles, future research directions should explore alternative strategies for preprocessing, the reduction of classifiers' computational costs, and word vectorization techniques. The concept can be expanded with the automatization of the conversion of incoming voice call recordings to text format and the development of a real-time routing system. The further classification for Turkish text advances, along with the expansion of NLP libraries, the better improved accuracy in text classification efforts will be. On the other hand, and in order to balance the number of records among the departments in our dataset, different sampling approaches can be explored.

Data Statement

Data sharing is not applicable to this article as it includes sensitive data regarding the Republic of Turkiye Ministry of Trade.

Acknowledgments

We would like to thank the General Directorate of Information Technologies of the Republic of Turkiye Ministry of Trade for generously providing access to the call center data for the purposes of this study.

Conflict of Interest

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

REFERENCES

[1] V. Mehrotra, T. A. Grossman, and D. A. Samuelson, "Call Center Management," *Wiley Encycl. Oper. Res. Manag. Sci.*, no. January, 2011, doi: 10.1002/9780470400531.eorms0130.

[2] A. O. Adeyemi, M. A. Saouli, and B. Sinha, "Influence of Call Centers on Emerging Business Models and Practices," *Asian J. Bus. Manag.*, vol. 6, no. 5, pp. 44–52, 2018, doi: 10.24203/ajbm.v6i5.5467.

[3] S. Chaudhary, N. Nasir, S. Ur Rahman, and S. Masood Sheikh, "Impact of Work Load and Stress in Call Center Employees: Evidence from Call Center Employees," *Pakistan J. Humanit. Soc. Sci.*, vol. 11, no. 1, pp. 160–171, 2023, doi: 10.52131/pjhss.2023.1101.0338.

[4] S. Ananthram, M. J. Xerri, S. T. T. Teo, and J. Connell, "High-performance work systems and employee outcomes in Indian call centres: a mediation approach," *Pers. Rev.*, vol. 47, no. 4, pp. 931–950, 2018, doi: 10.1108/pr-09-2016-0239.

[5] A. Keser and G. Yilmaz, "Workload , Burnout , and Job Satisfaction Among Call Center Employees," *J. Soc. Policy Conf.*, no. 66–67, pp. 1–13, 2014.

[6] J. Chatterjee, A. Saxena, and G. Vyas, "An automatic and robust system for identification of problematic call centre conversations," *Proc. - 2016 Int. Conf. Micro-Electronics Telecommun. Eng. ICMETE 2016*, pp. 325–330, 2016, doi: 10.1109/ICMETE.2016.48.

[7] G. Mishne, D. Carmel, R. Hoory, A. Roytman, and A. Soffer, "Automatic analysis of call-center conversations," *Int. Conf. Inf. Knowl. Manag. Proc.*, no. May 2014, pp. 453–459, 2005, doi: 10.1145/1099554.1099684.

[8] D. Galanis, S. Karabetsos, M. Koutsombogera, H. Papageorgiou, A. Esposito, and M. T. Riviello, "Classification of emotional speech units in call centre interactions," *4th IEEE Int. Conf. Cogn. Infocommunications, CogInfoCom 2013 - Proc.*, pp. 403–406, 2013, doi: 10.1109/CogInfoCom.2013.6719279.

[9] O. Rashid, A. M. Qamar, S. Khan, and S. Ambreen, “Intelligent decision making and planning for call center,” *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 1–6, 2019, doi: 10.1109/ICCISci.2019.8716472.
[10] I. O. Yigit, A. F. Ates, M. Guvercin, H. Ferhatosmanoglu, and B. Gedik, “Çağrı Merkezi Metin Madenciliği Yaklaşımı,” in *2017 25th Signal Processing and Communications Applications Conference, SIU 2017*, Institute of Electrical and Electronics Engineers Inc., Jun. 2017. doi: 10.1109/SIU.2017.7960138.
[11] K. Fiok *et al.*, “Text Guide: Improving the Quality of Long Text Classification by a Text Selection Method Based on Feature Importance,” *IEEE Access*, vol. 9, pp. 105439–105450, 2021, doi: 10.1109/ACCESS.2021.3099758.
[12] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, “A comparative study of automated legal text classification using random forests and deep learning,” *Inf. Process. Manag.*, vol. 59, no. 2, 2022, doi: 10.1016/j.ipm.2021.102798.
[13] M. Sarı, “Derin Öğrenme Yöntemleri Kullanılarak Türkçe Doküman Sınıflandırma,” TOBB University of Economics and Technology, 2018.
[14] K. Koruyan and A. EKERYILMAZ, “Makine Öğrenmesi ile Müşteri Şikayetlerinin Sınıflandırılması,” *AJIT-e Acad. J. Inf. Technol.*, vol. 13, no. 50, pp. 168–183, 2022, doi: 10.5824/ajite.2022.03.004.x.
[15] O. Uslu and S. Akyol, “Türkçe Haber Metinlerinin Makine Öğrenmesi Yöntemleri Kullanılarak Sınıflandırılması,” *Eskişehir Türk Dünyası Uygul. ve Araştırma Merk. Bilişim Derg.*, vol. 2, no. 1, pp. 15–20, Jan. 2021, Accessed: Dec. 16, 2022.
[16] B. Karakus, G. Aydin, and I. R. Hallac, “Distributed Readability Analysis of Turkish Elementary School Textbooks,” *Proc. Int. Conf. Inf. Technol. Comput. Sci.*, pp. 80–87, 2015.
[17] R. S. Kuzu, A. Haznedaroglu, and M. Levent Arslan, “Topic identification for Turkish call center records,” pp. 1–4, 2012, doi: 10.1109/siu.2012.6204647.
[18] H. Saif, M. Fernandez, Y. He, and H. Alani, “On stopwords, filtering and data sparsity for sentiment analysis of twitter,” *Proc. 9th Int. Conf. Lang. Resour. Eval. Lr. 2014*, no. i, pp. 810–817, 2014.
[19] G. Gupta, “Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example),” *Int. J. Comput. Appl.*, vol. 1, no. March 2015, pp. 60–768887, 2009.
[20] T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, “Stemming and lemmatization in the clustering of finnish text documents,” *Int. Conf. Inf. Knowl. Manag. Proc.*, no. May 2014, pp. 625–633, 2004, doi: 10.1145/1031171.1031285.
[21] A. Barbaresi, “simplemma,” 2022. http://doi.org/10.5281/zenodo.4673264
[22] A. K. Singh and M. Shashi, “Vectorization of text documents for identifying unifiable news articles,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 305–310, 2019, doi: 10.14569/ijacsa.2019.0100742.
[23] R. Zhao and K. Mao, “Fuzzy Bag-of-Words Model for Document Representation,” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, Apr. 2018, doi: 10.1109/TFUZZ.2017.2690222.
[24] W. Aljedaani *et al.*, “Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry,” *Knowledge-Based Syst.*, vol. 255, p. 109780, 2022, doi: 10.1016/j.knosys.2022.109780.
[25] S. Maleki, M. Musuvathi, and T. Mytkowicz, “Semantics-preserving parallelization of stochastic gradient descent,” *Proc. - 2018 IEEE 32nd Int. Parallel Distrib. Process. Symp. IPDPS 2018*, pp. 224–233, 2018, doi: 10.1109/IPDPS.2018.00032.
[26] M. Bhattacharya and D. Datta, “Diabetes Prediction using Logistic Regression and Rule Extraction from Decision Tree and Random Forest Classifiers,” *2023 4th Int. Conf. Emerg. Technol. INCET 2023*, pp. 1–7, 2023, doi: 10.1109/INCET57972.2023.10170270.
[27] V. Vijay and P. Verma, “Variants of Naïve Bayes Algorithm for Hate Speech Detection in Text Documents,” *2023 Int. Conf. Artif. Intell. Smart Commun. AISC 2023*, pp. 18–21, 2023, doi: 10.1109/AISC56616.2023.10085511.
[28] T. K. An and M. H. Kim, “A new Diverse AdaBoost classifier,” *Proc. - Int. Conf. Artif. Intell. Comput. Intell. AICI 2010*, vol. 1, pp. 359–363, 2010, doi: 10.1109/AICI.2010.82.
[29] M. A. Elgammal, H. Mostafa, K. N. Salama, and A. Nader Mohieldin, “A Comparison of Artificial Neural Network(ANN) and Support Vector Machine(SVM) Classifiers for Neural Seizure Detection,” *Midwest Symp. Circuits Syst.*, vol. 2019-Augus, pp. 646–649, 2019, doi: 10.1109/MWSCAS.2019.8884989.
[30] S. Akuma, T. Lubem, and I. T. Adom, “Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets,” *Int. J. Inf. Technol.*, vol.14, no.7, pp. 3629–3635, 2022, doi: 10.1007/s41870-022-01096-4.
[31] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: An overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000, doi: 10.1093/bioinformatics/16.5.412.
[32] B. C. ÖĞE and F. KAYAALP, “Farklı Sınıflandırma Algoritmaları ve Metin Temsil Yöntemlerinin Duygu Analizinde Performans Karşılaştırılması,” *Düzce Üniversitesi Bilim ve Teknol. Derg.*, vol. 9, pp. 406–416, 2021, doi: 10.29130/dubited.1015320.
[33] B. EKİCİ and H. TAKCI, “Spam Tespitinde Word2Vec ve TF-IDF Yöntemlerinin Karşılaştırılması ve Başarı Oranının Artırılması Üzerine Bir Çalışma,” *Bilecik Şeyh Edebali Üniversitesi Fen Bilim. Derg.*, vol. 8, no. 2, pp. 646–655, 2021, doi: 10.35193/bseufbd.935247.
[34] Ö. ÇELİK and B. C. KOÇ, “TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması,” *Deu Muhendis. Fak. Fen ve Muhendis.*, vol. 23, no. 67, pp. 121–127, 2021, doi: 10.21205/deufmd.2021236710.

APPENDIX (A)

TABLE A.1. The distribution of call records among the 99 department of RTMTCC and avarage number of words and characters in the call records of each department.

Department	The Number of Call Text	Avarage Number of Words	Avarage Number of Characters
Department 1	32316	34	253
Department 2	12896	39	316
Department 3	8919	24	171
Department 4	8627	31	261
Department 5	6747	31	238
Department 6	3452	26	183
Department 7	3449	27	189
Department 8	3327	25	168
Department 9	2649	43	312
Department 10	2483	34	273
Department 11	2185	29	220
Department 12	1865	26	175
Department 13	1377	28	217
Department 14	1061	25	205
Department 15	1018	70	542
Department 16	676	34	266
Department 17	629	61	515
Department 18	594	30	231
Department 19	389	43	326
Department 20	360	69	518
Department 21	240	37	282
Department 22	203	37	264
Department 23	203	31	251
Department 24	196	26	197
Department 25	195	33	242
Department 26	178	70	509
Department 27	177	32	234
Department 28	159	46	315
Department 29	155	146	1132
Department 30	137	129	991
Department 31	119	79	582
Department 32	98	35	256
Department 33	93	48	386
Department 34	83	33	237
Department 35	81	41	302
Department 36	76	36	298
Department 37	69	31	227

Department 38	59	29	237
Department 39	50	123	955
Department 40	46	25	184
Department 41	44	70	520
Department 42	40	44	309
Department 43	40	45	356
Department 44	40	76	584
Department 45	40	37	273
Department 46	40	77	581
Department 47	40	49	344
Department 48	40	137	1101
Department 49	40	47	337
Department 50	40	34	254
Department 51	40	89	690
Department 52	40	108	819
Department 53	40	25	194
Department 54	40	100	806
Department 55	40	32	253
Department 56	40	125	1021
Department 57	40	82	613
Department 58	40	30	203
Department 59	40	22	156
Department 60	40	24	209
Department 61	40	29	232
Department 62	40	133	1093
Department 63	40	145	1066
Department 64	40	109	831
Department 65	40	219	1720
Department 66	40	214	1654
Department 67	40	31	258
Department 68	40	67	523
Department 69	40	22	190
Department 70	40	49	363
Department 71	40	56	452
Department 72	40	81	664
Department 73	40	76	744
Department 74	40	33	245
Department 75	40	21	181
Department 76	40	35	256
Department 77	40	24	144
Department 78	40	22	202
Department 79	40	56	428
Department 80	40	56	396

Department 81	40	30	206
Department 82	40	36	248
Department 83	40	44	336
Department 84	40	20	180
Department 85	40	24	164
Department 86	40	20	144
Department 87	40	176	1544
Department 88	40	28	232
Department 89	40	32	228
Department 90	40	36	244
Department 91	40	24	188
Department 92	40	24	164
Department 93	40	16	172
Department 94	40	48	308
Department 95	40	28	200
Department 96	40	232	1788
Department 97	40	24	152
Department 98	40	24	188
Department 99	40	32	264