applied sciences

MDPI

Article

Machine Learning-Based Text Classification Comparison: Turkish Language Context

Yehia Ibrahim Alzoubi [1], Ahmet E. Topcu [2,*] and Ahmed Enis Erkaya [3]

[1] College of Business Administration, American University of the Middle East, Egaila 54200, Kuwait; yehia.alzoubi@aum.edu.kw
[2] College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait
[3] Tubıtak Bılgem Software Technologies Research Institute, Ankara 06100, Türkiye; aeniserkaya@gmail.com
* Correspondence: ahmet.topcu@aum.edu.kw

Abstract: The growth in textual data associated with the increased usage of online services and the simplicity of having access to these data has resulted in a rise in the number of text classification research papers. Text classification has a significant influence on several domains such as news categorization, the detection of spam content, and sentiment analysis. The classification of Turkish text is the focus of this work since only a few studies have been conducted in this context. We utilize data obtained from customers' inquiries that come to an institution to evaluate the proposed techniques. Classes are assigned to such inquiries specified in the institution's internal procedures. The Support Vector Machine, Naïve Bayes, Long Term-Short Memory, Random Forest, and Logistic Regression algorithms were used to classify the data. The performance of the various techniques was then analyzed after and before data preparation, and the results were compared. The Long Term-Short Memory technique demonstrated superior effectiveness in terms of accuracy, achieving an 84% accuracy rate, surpassing the best accuracy record of traditional techniques, which was 78% accuracy for the Support Vector Machine technique. The techniques performed better once the number of categories in the dataset was reduced. Moreover, the findings show that data preparation and coherence between the classes' number and the number of training sets are significant variables influencing the techniques' performance. The findings of this study and the text classification technique utilized may be applied to data in dialects other than Turkish.

Keywords: Turkish texts; machine learning; text preprocessing; algorithm effectiveness

check for updates

Citation: Alzoubi, Y.I.; Topcu, A.E.; Erkaya, A.E. Machine Learning-Based Text Classification Comparison: Turkish Language Context. *Appl. Sci.* **2023**, *13*, 9428. https://doi.org/10.3390/app13169428

Received: 4 July 2023
Revised: 31 July 2023
Accepted: 17 August 2023
Published: 19 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Corporations provide a variety of online applications using which consumers can submit grievances to relevant divisions. Daily, lots of complaints or inquiries are delivered to large corporations. It is critical for people that these inquiries be received and replied to. Individuals that make these inquiries desire for them to be met as quickly as is feasible. It wastes a lot of time for large corporations to send these inquiries to the right divisions, and the rate of inquiries lowers proportionately. The misclassification of the arriving inquiries even lengthens the response time [1]. The advancement of Machine Learning (ML) algorithms has resulted in remedy recommendations for our everyday difficulties [2]. ML deployment has become unavoidable for large corporations to adapt to meet customer inquiries [3].

ML often gives systems the capability to learn and improve based on experience without being explicitly coded [4]. Unsupervised, supervised, semi-supervised, and reinforcement learning are the four main themes of ML algorithms [5]. ML trains on data and discovers how to fulfill jobs using different algorithms. These algorithms try to extract secret knowledge from enormous amounts of available data and apply it to classification or regression models [6]. Therefore, ML may be of great assistance in text classification [7]. The

Support Vector Machine (SVM), Naïve Bayes (NB), Long Term-Short Memory (LTSM), Random Forest (RF), and Logistic Regression (LR) algorithms are some of the ML techniques [8], and these were deployed in this study. Section 3 will go through these algorithms in further depth. The ML workflow typically includes three steps: data preparation, selecting the appropriate ML algorithms and variables, and evaluating and assessing performance [9].

Many studies on the use of ML for text classification have recently been published, including survey papers (e.g., [2,10–12]), comparative analyses using different ML algorithms (e.g., [13,14]), papers focusing on specific languages (e.g., [4,15,16]), and papers applying certain ML algorithms for text classification (e.g., [1,17,18]), text classification performance (e.g., [19]), and text classification frameworks (e.g., [6]). However, the literature on text classification in the Turkish context is limited [20,21]. Accordingly, this work tries to meet this demand by focusing on several ML algorithms for identifying Turkish texts. It is intended that the inquiry text will be sent to the relevant class quickly using ML algorithms, allowing the staff to react to letters more quickly. Another concern to consider is which ML classification algorithm is best for this task. The classification method is challenging in this study as all of the inquiries are in the Turkish composable language [22,23].

The paper contributes in the following ways. We utilize data from customer inquiries that come to an institution to evaluate the proposed techniques. Classes are assigned to such inquiries, which are specified in the institution's internal procedures. This study utilized the customers' inquiries, which involved many spelling errors. Accordingly, to prepare these data for analysis, we first normalized them. The terms in the corpus were then morphologically analyzed into their basic forms. Furthermore, the stop word list was generated by examining the most repeated word groupings and removing them. Also, the repeated classes were simplified using the k-means technique during the preparation phase, and the number of classes was reduced. Therefore, the dataset became more accurate and equitable. The SVM, NB, LTSM, RF, and LR methods were used for data classification. The performance of the various techniques was then analyzed after and before data preparation.

The LTSM was found to be the most effective technique in terms of accuracy. Furthermore, the techniques used performed better once the categories in the dataset were reduced. Moreover, the findings show that normalized data and coherence between the categories' number and the number of training sets are significant variables of the techniques' performance. The comparison also demonstrates that the preparation stages have a big influence on the results. The Term Frequency–Inverse Document Frequency (TF-IDF) approach utilized in the feature extraction stage, on the other hand, has the biggest impact on the results. The rest of this study is organized as follows. Section 2 discusses the evolution of Natural Language Processing (NLP), the ML approach, and related literature. Section 3 discusses the proposed model structure and tools used to conduct this model. Section 4 discusses the data preparation process and explores the dataset before and after data preparation. Section 5 presents the findings of this study and the performance evaluation of the proposed model. Section 6 concludes this paper, outlines the next work of the research, and closes this study.

2. Literature Review

This section reviews all of the available literature on Turkish text classification because the focus of this study is on Turkish text classification. This section also provides the background of the NLP.

2.1. Text Classification in Turkish Context

Amasyalı and Diri [24] investigated the SVM, NB, and RF algorithms in terms of the document type, name of the author, and author's race. For text classification, they employed the character N-grams approach. The writings of 18 authors, 4 females and 14 males, with 35 distinct texts authored by every author, and common interests and athletics, were analyzed. The correlation-based feature selection approach was utilized,

which was found to improve classification accuracy. The best functioning method in author discovery was found to be the NB method, and the best working algorithm in race and genre identification was found to be the SVM [24]. Güran et al. [25] used DT-J48, K-Nearest Neighbor (K-NN), and Bayesian Probabilistic classifiers to classify text using the N-gram method. They looked into 600 text documents that were obtained online. These documents were represented by the researchers using TF-IDF. They also used stemmed words on the data to minimize the vector space. They changed the text to lowercase in data preparation, then applied data cleansing to create and stop the word lists. Later, they used the information gain approach to perform feature extraction and the splitting of words. All three techniques achieved the highest performance for unigram words. Among them, the K-NN was shown to be the lowest-performing ML algorithm compared to the others (i.e., the K-NN's performance was 65.5%, that of the BN was 94%, and 75% was the performance of J48). According to the authors, the roundness of the training dataset has a detrimental impact on the findings [25].

Uysal and Gunal [26] investigated the effect of data preparation on text classification in news and e-mail data in English and Turkish. They accomplished the preparation through text lowercasing, stop-word deletion, tagging, and breaking. According to them, the classification includes different phases of classification and the extraction and selection of features. According to the study, English represents a non-composable language, whereas Turkish is an example of a composable language that is widely used across the globe. The text was classified using the SVM and measured using Micro-F1 algorithms. They concluded that the preparation phase is just as critical as the processes of extraction and selection of features [26]. Furthermore, they stated that stop-word cleansing is a critical phase also. This research also contends that while some preparation operations, such as lowercase modification, are required irrespective of the language or field, others must be integrated according to the language and field of the study [26].

Yıldırım and Yıldız [27] compared the standard bag-of-words approach and artificial neural system for text classification. One of the datasets included seven distinct types of data such as sports, policy, and economics. The second dataset consisted of six distinct types, each with 600 texts. Using cleaning and stop-word procedures, they prepared the data for morphological analysis. As a classification algorithm, the NB was utilized [27]. Their findings demonstrated that stop-word cleansing and morphological analysis had little effect on the outcome. They also emphasized the significance of selecting features in the classification process and employing the Chi-Square and Knowledge Gain techniques [27]. Kuyumcu [28] used quick text classification (fastText) without data preparation since data preparation is frequently time-consuming, especially in composable languages such as Turkish. Facebook's fastText word embedding-based analyzer eliminates the requirement for data preparation. The fastText algorithm was used on the Turkish text classification 3600 dataset. They tested the model using the NB, K-NN, and J48 techniques. According to the results, the best performance was attained by the Multinomial NB classifier, which scored 90.12%. Accordingly, the author concluded that the fastText algorithm is substantially superior compared to other methods in terms of consistency without the need for data preparation [27].

Çoban et al. [29] used Deep Learning (DL) to perform sentiment analysis on public Facebook data acquired from Turkish user profiles. With text representations, recurrent neural systems obtained the highest accuracy, 91.6%. Dogru et al. [30] suggested a DL-based classification of news texts using the Doc2vec word-based approach on the Turkish text classification 3600 dataset. DL-based Convolutional Neural Networks (CNN) and classic ML methods such as the NB, SVM, RF, and Gauss NB algorithms were employed as classification techniques. In the suggested model, the maximum result was reached as 94.17% in the Turkish sample compared to 96.41% in the English sample in the classifications performed by CNN [30]. Zulqarnain et al. [31] employed DL to perform question classification in Turkish. They used three different DL algorithms (Gated Recurrent Unit, LSTM, and CNN). They also used the word2vec methodology. Word2vec strategies had

a major effect on the prediction performance utilizing multiple DL algorithms, which achieved an accuracy of 93.7% [31].

Bektaş [32] used text classification tools to analyze 7731 tweets from 13 prominent Turkish economists. The classification findings were then compared to four popular ML algorithms (the SVM, NB, LR, and integration of LR with the SVM). The results revealed that the success of a text classification issue is related to the feature extraction techniques and that the SVM outperforms other ML methods using unigram feature maps. The integration approach of the SVM with LR generated the best results (82.9%) [32]. Bozyigit et al. [23] deployed ML to classify customers' concerns regarding packaged food goods expressed in Turkish. The class of concern was determined using the TF-IDF and word2vec feature extraction algorithms. The results of the LR, NB, K-NN, SVM, RF, and Extreme Gradient Boosting classifiers were compared. The strongest technique was Extreme Gradient Boosting with an TF-IDF weighted value, which reaches an 86% F-measure score [23]. When compared to the TF-IDF method, word2vec-based ML performed poorly in terms of the F-measure. Furthermore, TF-IDF-based ML provides more accurate predictions on the optimized feature subsets determined by the Chi-Square approach, which, when conducted on TF-IDF features, raises the F-measure from 86% to 88% in Extreme Gradient Boosting [23].

Eminagaoglu [33] introduced a similarity metric for classification that can be utilized well for word vectors and classification techniques such as the K-NN and k-means methods. The suggested metric is validated against certain global datasets in English and Turkish. The suggested metric might be employed in any applicable method or model for data acquisition and text classification. Karasoy and Ballı [22] performed a content-based SMS classification utilizing ML and DL approaches to filter out undesirable texts in Turkish. The features were analyzed using DL and ML and the results were compared. As a consequence, the CNN technique was discovered to be the best successful technique, with a 99.86% accuracy rate in classification.

Köksal and Yılmaz [21] proposed a technique and considerations for improving text categorization effectiveness with ML. ML methods and state-of-the-art preparation text models were used to assess two publicly available Turkish news datasets. They used a variety of ML techniques, including the NB, LR, K-NN, SVM, and RF techniques. They also used a BERT model that was particularly trained for classifying Turkish text. The results demonstrated that the technique outperformed earlier F1-score-based news classification experiments and achieved 96% accuracy [21]. Yildiz [20] presented a data distribution algorithm that addresses the data imbalance issue to improve text categorization success. The suggested algorithm was evaluated using LSTM on a very large Turkish dataset having 263168 articles divided into 15 groups. To compare the parameters, the model was trained with and without utilizing the suggested algorithm. The proposed algorithm produced roughly 3.5% more accuracy than the standard approach experiment. It also revealed a more-than-three-point rise in the F1-score [20]. Table 1 summarizes the findings of the previous studies.

This study prioritized data preparation by normalizing the data, morphologically analyzing the terms in the corpus into their basic forms, generating the stop words' list by examining and removing the most repeated word groupings, simplifying the repeated categories using the k-means technique during the preparation phase, and reducing the number of categories. The dataset was trained using the SVM, NB, LTSM, RF, and LR methods. This study can be assumed to be an extension and validation of [21–23]. The effectiveness of the various strategies was then evaluated both after and before data preparation. It is also important to note that this article is a part of and is based on [34].

Table 1. Recent studies on ML Turkish text classification.

Study	Technique/Algorithm Used	Findings
[24]	SVM, NB, RF	The NB is the best for author discovery; the SVN is the best for race and genre identification
[25]	DT-J48, K-NN, Bayesian Probabilistic classifiers, N-gram method	K-NN performance achieved 65.5%, BN achieved 94%, and J48 achieved 75%
[26]	SVM, Micro-F1	The preparation phase is just as critical as the processes of the extraction and selection of features
[27]	Bag-of-words approach, artificial neural system	Stop-word cleansing and morphological analysis had little effect on the outcome
[28]	FastText tested using NB, K-NN, J48	Multinomial NB classifier achieved the best at 90.12%
[29]	DL techniques	Recurrent neural systems obtained the highest accuracy of 91.6%
[30]	DL, ML techniques (NB, SVM, RF, and Gauss NB)	94.17% in the Turkish sample compared to 96.41% in the English sample in the classifications performed by CNN
[31]	DL techniques (Gated Recurrent Unit, LSTM, CNN)	DL algorithms achieved an accuracy of 93.7% on the question dataset
[32]	SVM, NB, LR, and integration LR with SVM	The integration approach of the SVM with LR generated the best results (82.9%)
[23]	LR, NB, K-NN, SVM, RF applied on TF-IDF and word2vec	Extreme Gradient Boosting with an TF-IDF weighted value scored the best (86%) F-measure score
[33]	Proposed a similarity metric that can be used for K-NN and k-means	The suggested metric might be employed in any applicable method or model for data acquisition and text classification
[22]	ML (NB, RF, SVM, multilayer perceptron, Random Subspace, LR, K-NN, DL (CNN and LSTM)	CNN scored the best with a 99.86% accuracy rate
[21]	NB, LR, K-NN, SVM, RF	The new technique outperformed earlier F1-score-based news classification experiments and achieved 96.00% accuracy.
[20]	LSTM	A new data distribution methodology
This study	k-means technique, TF-IDF, SVM, NB, LTSM, RF, LR	LTSM was found to be the most effective technique in terms of accuracy, and data preparation is important for the overall performance of the algorithm used

2.2. Natural Language Processing

NLP is an AI discipline that deals with the processing of human language in a computer-readable format. NLP, which was designed to help computers comprehend the language humans use to communicate, has become very common [13]. The ease with which people's speech may now be accessed via social media and the proliferation of communication outlets such as radio and television have enlarged the usage of NLP. NLP is required in text classification to analyze user data. Building a process by making meaning out of text data is a difficult step. NLP is practiced in several stages. The sentences, firstly, are divided into smaller sections called tokens in lexical analysis. The structure of the sentence or syntax of all tokens is then considered in the syntactic analysis process [12]. It is tested to see whether its syntax is correct. The interpretations of phrases are inferred

using the preceding procedures in semantic analysis. The NLP stages are completed after turning the data into output data.

In this work, a morphology analysis tool, a key element of NLP, was employed in data preparation. Turkish morphology analysis was carried out using the Zemberek library. Zemberek performs parsing with root dictionary-based parses. The library conducts parsing by calculating the probability of the root. The library initially scans the binary root record stored before appending the relevant sentences to the root. The origin of a term in Turkish can take several forms. Various changes were added to the tree for such assertions and distorted terms. The Turkish term "kitaba", for instance, is different from the term "kitap", which is the origin of the term. As a result, the distorted term "kitab" was added to the tree [35].

TRMOR is yet another morphological technique for Turkish analysis, developed by Kayabas et al. in [36]. TRMOR's performance was evaluated using 1000 words chosen at random from Wikipedia. TRMOR achieved an accuracy value of 94.12%. TRMOR binds stems to suffix morphemes first, assessing all possible connections, and then maps the output string in the right surface form using morphological criteria [36]. Since Turkish is an agglutinative (composable) language, composite terms are difficult to parse. For example, while "su" is the compound marker in "acemborusu", "i" is the composite marker in "ayçiçeği". According to the study, every composite term currently cannot be addressed, although this may be achievable in future projects [36].

2.3. Machine Learning

ML refers to teaching a set of algorithms to a machine (computer) using data, where there are no set rules [12]. The machine learns by finding patterns or commonalities in the dataset. ML algorithms cannot be employed when there are no patterns. The machine does its own data parsing to determine what action to take. As humans have to practice anything new, we learn frequently, and a computer's data quantity is connected to a more efficient learning process [2]. Data are essential in ML, and it has gotten simpler to obtain data as the internet has improved, increasing the popularity of ML. According to the approaches utilized, ML algorithms are classified as unsupervised or supervised [22]. As mentioned in the introduction, several supervised algorithms were utilized in this work, including the K-NN, LR, SVM, RF, and NB algorithms. Furthermore, the LTSM method is employed as an unsupervised DL algorithm [2].

2.3.1. Supervised Learning

It is a type of ML activity wherein "supervised" refers to data that have been labeled [13]. A sample dataset is used by the ML model. The primary objective is to obtain the description of the data class. Algorithms for supervised learning discover the connections and correlations between output and input and anticipate the next output. That is, the data to be utilized in supervised learning and the classes associated with it are linked. They are known as labeled data [10]. Classification is a systematic method for generating classifications utilizing the prediction of training examples based on previous data labels. That is why this method is known as supervised learning. Examples of these classifiers include a Rule-Based classifier, DT classifier, Neural Network classifier, Neuro-Fuzzy classifier, the SVM, and so on. In this paper, the SVM classifier, NB classifier, LR classifier, and RF classifier were utilized using labeled textual data.

2.3.2. Unsupervised Learning

In unsupervised learning, computers execute the process of learning by detecting patterns in the data. There is no particular output, or associated data, in the dataset (i.e., there are no data labels). The computer builds models by looking for similarities and patterns in the inputs [23]. Unsupervised ML seeks a relationship among datasets, which might be positive or negative. In other words, unsupervised ML discovers patterns of similarities or distinctions across datasets. Because there is no sample organizer or dataset, this method is referred to as unsupervised learning [6]. Unsupervised ML methods that

have been utilized include the Hidden Markov Model, k-means Clustering, LTSM method, and Singular-Value Decomposition. This study employs both the k-means algorithm as well as the LTSM method, which is also regarded as a DL algorithm [12].

3. Research Method

The architecture of the suggested model, the algorithms, and the technologies utilized to build the model in this study are discussed in this section. There are several methods used for performing text classification. The procedure employed in this study is depicted in Figure 1. Since supervised ML algorithms are used, we require labeled documents to begin text classification. After preparation, the data must go through many steps for the algorithm to function properly. The results are then compared among the different algorithms used regarding time of training and accuracy. In addition, the results obtained after and before using ML algorithms are compared to assess the efficacy of these algorithms.

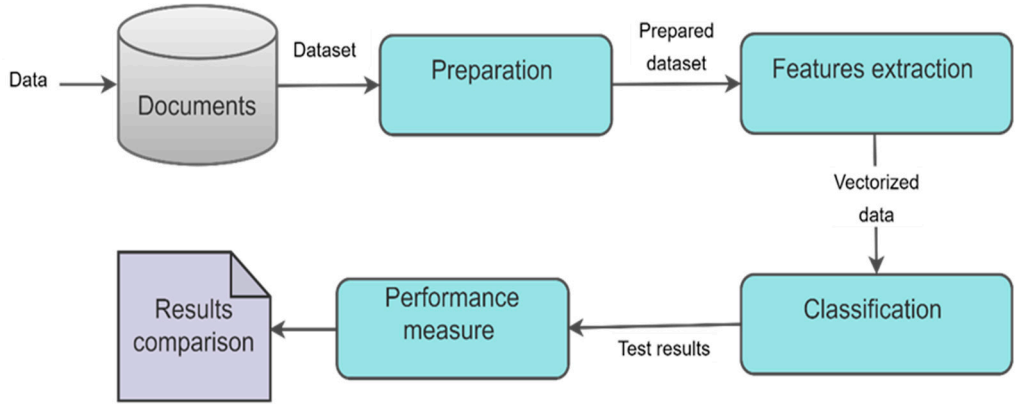


Figure 1. Classification process used in this study.

The algorithms utilized in this study are described in this section, as indicated in Section 2. The proposed model was created using several algorithms including the K-NN algorithm, SVM classifier, NB classifier, LR classifier, RF classifier, and the LTSM method. The following sections describes the tools necessary to build and run these algorithms.

3.1. Python Tool

The Python programming language and libraries were utilized to carry out the data classification. Python is a well-known tool in computer science. We employed five Python libraries in this study, Pandas, Matplotlib, NumPy, Scikit-learn, and Keras, which are presented in the following paragraph.

Pandas' library is widely used in data science. It may be utilized to transform a dictionary, Python list, or NumPy array into a Pandas dataset for data analysis [37]. Matplotlib is a graphical interface toolkit that includes several functions such as creating visual datasets. It is feasible to show data with a smaller number of lines of code [38]. Scikit-learn contains several unsupervised and supervised ML techniques [39]. The Scikit-learn Library was utilized to construct all of the algorithms in this research. Keras may be run on GPU or CPU. If it is executed on GPU, the training time may be significantly reduced [40]. A sequential model, for example, is downloaded from Keras' models using the line of code "from keras.models import Sequential". Since the LSTM algorithm had to be created in tiers, we employed this form of the model in our research.

3.2. Zemberek

The Zemberek [41] library is an open source for phrases in Turkish. Normalization, tokenization, language modeling, morphology, language identification, named entity recognition, the GRPC server, and classification are all modules in Zemberek. Word synthesis, Turkish morphological analysis, and disambiguation may all be performed with the Mor-

phology module. Sentence boundary detection and tokenization may be employed via the Tokenization module. Word suggestion, simple spell checking, and noisy text normalization may all be utilized using the Normalization module. Turkish named entity recognition may be utilized with the Named Entity Recognition module. Text data may be categorized using the Classification module. The Morphology and Normalization modules are employed in this investigation.

4. Data Preparation

The dataset utilized to classify the data in this study is detailed in this section. The goal of classifying data is to determine which group the texts in a corpus belong to [42] such that if {c1, c2, . . ., cn} is the collection of all classes and "di" is a text from the whole set of documents D, text classification gives one category "cj" to a document "di". The text in consideration may be classified into only one or more classes. For example, a magazine article may be classified as belonging to the health and sports classes. When the text is linked to only one category, it is referred to as "single-label"; if it is associated with more than one class, it is referred to as "multi-label". The findings were analyzed by employing several algorithms to determine with which class the inquiry of request received by an organization is associated. This section also includes information about the word structure and the number of trigrams (a trigram is an n-gram with n = 3, meaning that it consists of three adjacent words or tokens from a text), bigrams (a bigram is an n-gram with n = 2, meaning that it consists of two adjacent words or tokens from a text), and unigrams (unigram is an n-gram with n = 1, meaning that it consists of a single word or token from a given text) in the dataset before and after preparation.

4.1. Data Preparation Steps

Figure 2 depicts the data preparation stages. Data preparation is essential to train the texts consistently using ML algorithms. If the data are obtained directly from consumers, they may contain spelling mistakes and superfluous letters or phrases. These spelling mistakes may cause the classification algorithms to perform incorrectly. Since these algorithms treat multiple spellings of the same word as separate terms, the results may change [1]. The data cleaning process commences by converting all texts to lowercase.

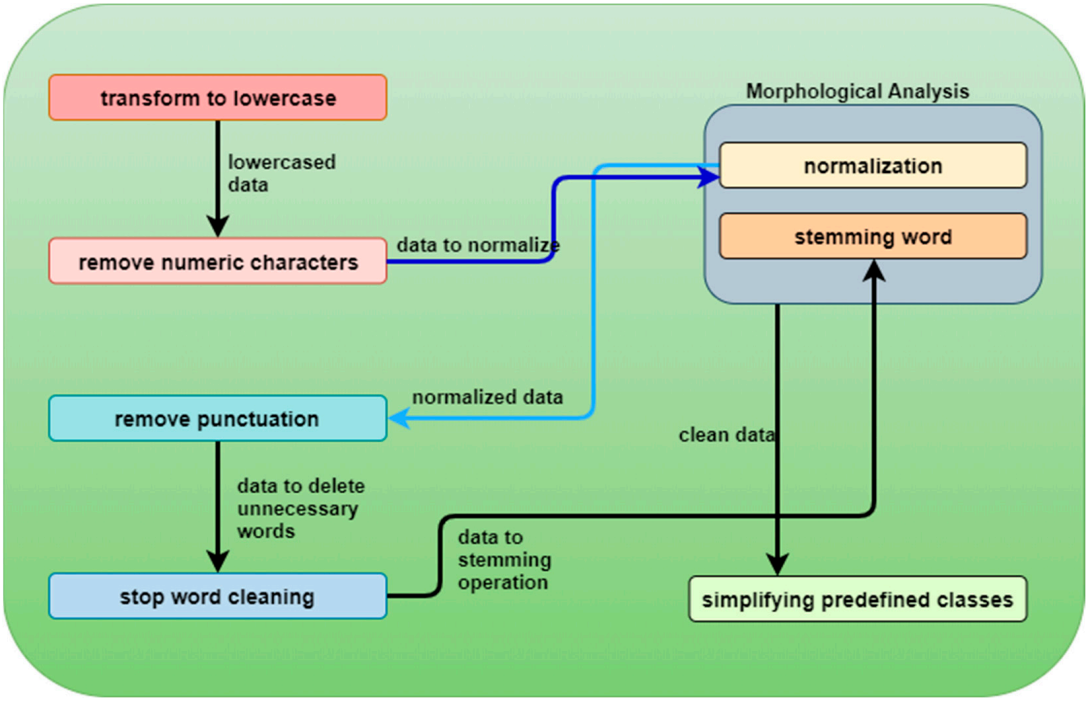


Figure 2. Data preparation steps [34].

The text's special characters, digits, and punctuation marks are then removed. Finally, the stop words are eliminated. These stages are essential for the data to perform more consistently and accurately. Because inquiries are created by users, the terms included inside them may be incorrect or inappropriate. These words are rectified using normalization. The data are then deduplicated across all documents using the steaming process. In NLP, these stages are known as morphological analysis [36]. In this study, the numerical letters were then eliminated, followed by normalization as a morphological analysis phase. Finally, the data were processed to excerpt feature vectors using the TF-IDF technique. The data vectorized using the TF-IDF technique were classified using the NB, SVM, RF, LSTM, and LR algorithms, and the findings were contrasted.

4.2. Data Exploration

The data utilized in this research came from a corporation in Turkey that receives hundreds of inquiries daily. There are 225,239 consumer inquiries and 1819 themes in the dataset. Table 2 compares the number of terms before preparation to the number of terms after data preparation. A customer's inquiry is roughly 33 words in length before preparation. The inquiries must have at least one and no more than 3026 words. Table 2 also provides data regarding the number of terms in the corpus texts after preparation. A document is made up of about 18 terms. It is also mentioned that the inquiry must have at least one and no more than 1171 words. It can be observed that preparation deleted about 15 terms from each inquiry.

Table 2. Comparing the number of terms before and after data preparation.

Dataset	Maximum Number of Words	Minimum Number of Words	Average Number of Words
Raw	3026	1	33
Prepared	1171	1	18

The efficiency of the algorithms employed in this study was evaluated based on their training time, accuracy, recall, and precision. The experiment was conducted on a system equipped with an Intel® CoreTM i7-6820HQ 2.70 GHz CPU and 32 GB of RAM. Using this system, the algorithms generated their respective models. Notably, the training times for each algorithm varied across the three distinct assertions present in the provided dataset. The training times for each method for the dataset's three separate assertions are presented in Table 2 below in seconds. It is evident from the table that all algorithms necessitated a considerable training period when processing the raw information. However, the preprocessed dataset yielded the fastest training times among all methods.

According to the table, the NB is the algorithm with the shortest training time among the methods used. The learning processes of the algorithms depend on computations, iteration counts, and class counts, and they are all directly proportional. As a result, the presence of dataset imbalances and an increasing number of classes can lead to longer learning times. Similar to what occurs in humans, machine learning also takes longer as the number of classes increases. The complexity of the calculations also impacts the duration of the learning process for the algorithms.

The study obtained its data from a metropolitan municipality located in Turkey. This municipality comprises various departments, each responsible for handling specific types of applications. For example, if someone submits an application concerning an injured cat, it gets directed to the veterinary department. Similarly, applications related to issues such as flooded apartments are forwarded to the water and sewerage department. In this dataset, one can observe the allocation of each application to its corresponding department.

The applications in the study were already labeled with their respective classes. Figure 3 compares the pre- and post-preprocessing states of the applications. Notably, removing conjunctions from the application text revealed meaningful data. In Figure 3b,

one notes that words related to topics of interest to the municipality, such as 'water', 'work', 'bus station', 'bus route', 'Çankaya' (district name), 'pavement', and 'dog', emerged.

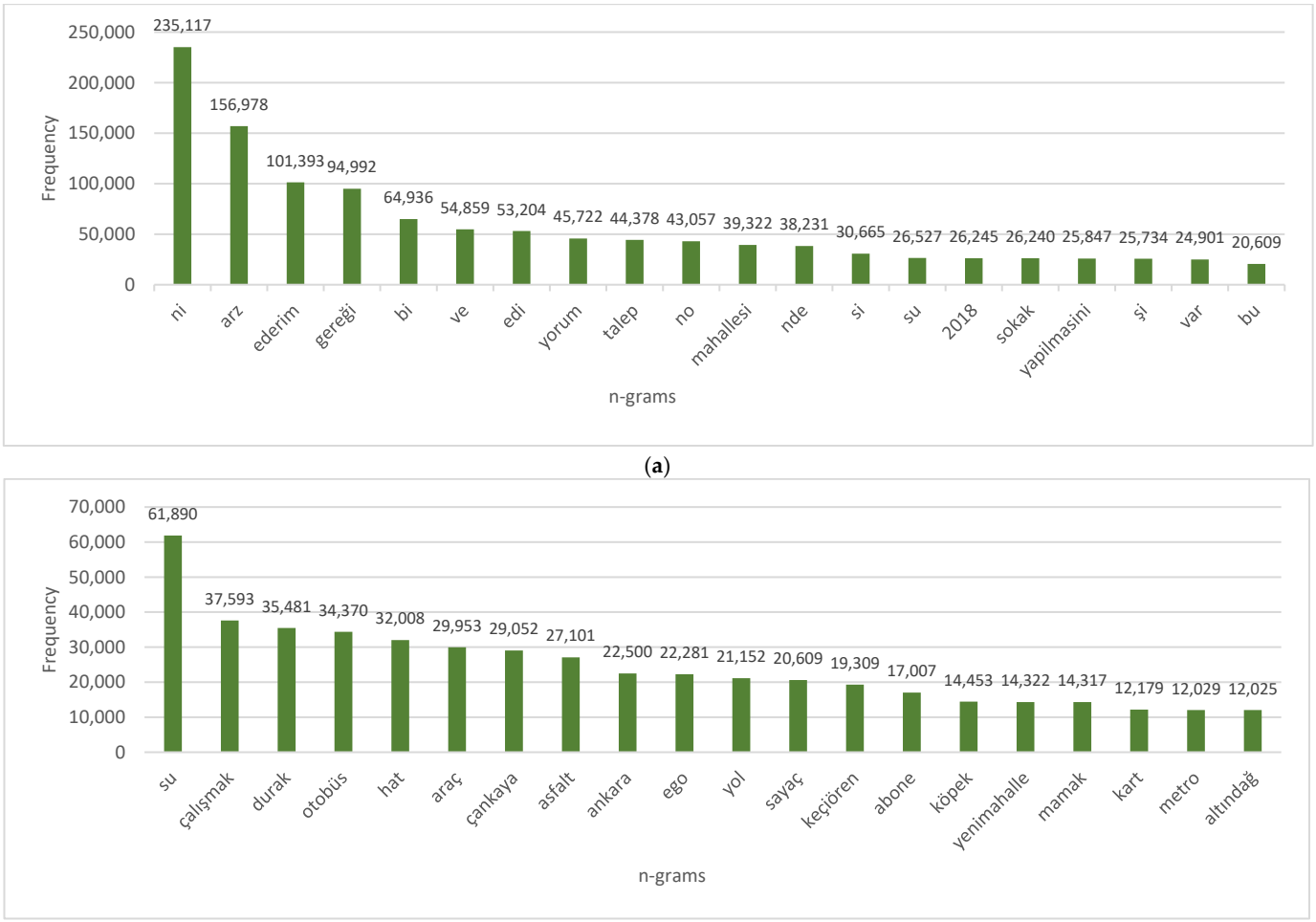
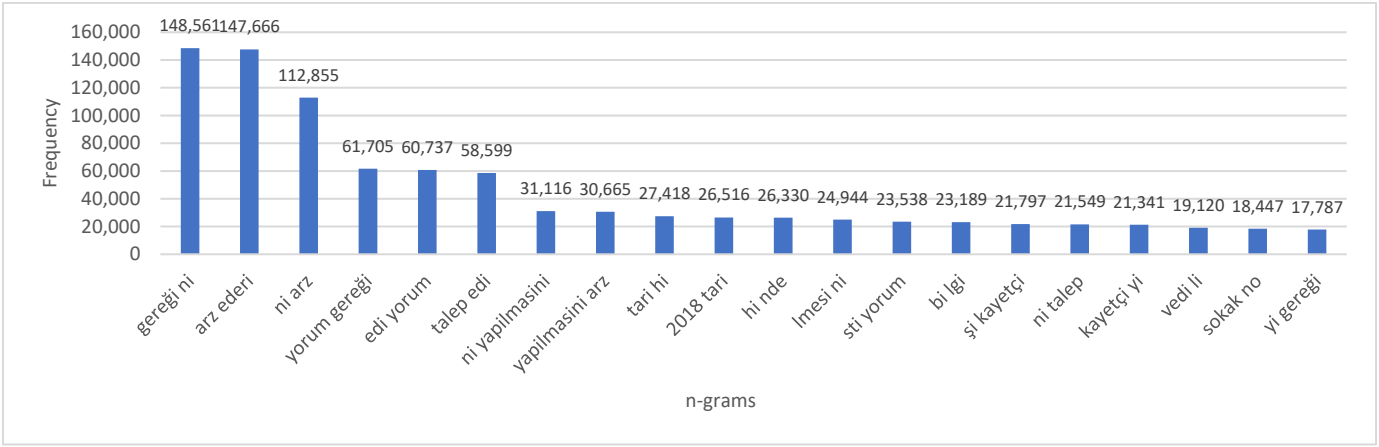


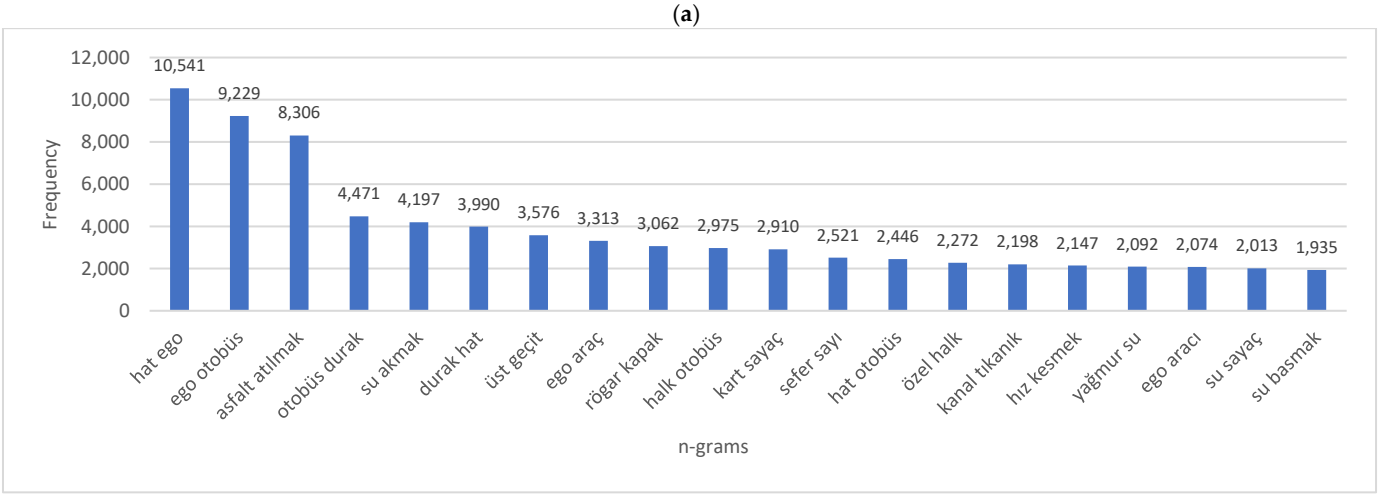
Figure 3. Unigram data comparison: (**a**) most commonly occurring unigrams before preparation; (**b**) most commonly occurring unigrams after preparation.

Figure 4 lists bigrams, but those in (a) were found to be irrelevant for topic detection as they merely express a common request sentence found at the end of every application. However, after preprocessing, certain bigrams such as 'hat ego' became meaningful, signifying 'bus route'. Other word groups were also related to the municipality's areas of interest. For instance, "kanal tıkanıklığı" means 'sewer blockage' in English, showing words relevant to the municipality's field of work.

The dataset underwent training using machine learning algorithms to generate models both before and after preprocessing. The test data comprised 25% of the dataset, containing 225,239 letters of request, while the remaining data served as the training dataset. The RF algorithm stands out as an ensemble method, resembling a forest with multiple decision trees. For the RF classifier in Scikit-learn, the parameter specifying the number of decision trees was set to 200 and the maximum depth was set at three. Furthermore, models were created using the SVM, NB, and LR algorithms, all present in Scikit-learn. In the case of the SVM, the linear kernel was applied. Subsequently, the generated models were tested using the designated test data, and their precision, recall, F1-score, and accuracy values were compared against each other. The training times for the five algorithms—SVM, RF, NB, LR, and LSTM—are compiled in Table 3.



(**a**)



(**b**)

Figure 4. Most commonly occurring bigrams before and after data preparation: (**a**) most commonly occurring bigrams before preparation; (**b**) most commonly occurring bigrams after preparation.

Table 3. Comparing training time.

Dataset	SVM	RF	NB	LR	LSTM
Raw	46,943.523	36.727	41.663	5850.307	60,214.325
Preprocessed without simplification step	9879.506	23.387	20.232	5418.467	58,879.152
Preprocessed	8865.088	7.445	0.227	39.614	10,426.658

4.2.1. Stop Word List

Some terms such as “gerei”, and “talep” are frequently used since these data were obtained from an organization’s inquiries. Terms including “yaplmasn talep ederim”, “gereini arz ederim”, “sayn”, and “başkan” were all cleaned throughout the preparation stage as stop words. The stop word list used in this study is shown in Table 4.

Table 4. Comparing the number of the terms before and after data preparation.

Stop word	sokak, cadde, mahalle, mah, mh, istinaden, no, tel, cep, sk, faks, te, kap, iç, gerek, bulvar, ilçe, il, arz, sayın, etmek, başvuru, eski, meydan, gelmek, null, saat, fax, cad, sok, ara, civar, bura, ora, kişi, görev, başlamak, yaşamak, binmek, sıkıntı, ad, taraf, soy, acilen, çöz, bulunmak, müdahale, numara, bilgi, vermek, birim, söz, yarmak, iyi, sayın, günlemek, tarih, yetkili, başkan, mağdur, vatandaş, şikayet, nol, anmak, yeni, ivedilikle, mağdur, temiz, yolmak, zor, kalmak, demek, almak, bina, gitmek, patlak, konu, ev, durum, istemek, kontrol, geçmek, nol, ivedi, rica, mevcut, park, gün, site, kullanmak, büyükşehir, bey, beklemek, lütfen, yok, mağduriyet, gidermek, talep, şikâyet, belediye.

4.2.2. Unigram Data Comparison

When the raw data are analyzed, the most repeating 20 unigrams are presented in Figure 3a. When the word clusters and the most repeated terms are examined, the corpus contains a large number of useless terms. For example, the spelling "ni" is replicated over 200,000 times, which is meaningless in Turkish. Because of the users' spelling mistakes, such typos are incorporated into the inquiries. Furthermore, due to citizen spelling errors, some of the terms in the bigrams appear to be nonsensical.

Figure 3b depicts the 20 most repeating unigram terms after preparation. Sensible terms such as "su", "durak", and "çalşmak" are some of the most repeated bigrams and unigrams after preparation. The words used as stop words with a greater incidence before preparation were eliminated during the preparation stages. When comparing Figures 3a and 4b, it is clear that considerable data cleansing was accomplished using stop-word filtering. When looking at the most recurrent unigram terms in Figure 3b, it is clear that the keywords obtained after preparation are more reliable. Terms such as "su", "çalşmak", "durak", and "hat" are some of the functioning domains of the corporation and will aid in the detection of linked classes. When compared to Figure 3a, the unigrams obtained after preparation are more reliable and intelligible.

4.2.3. Bigram Data Comparison

When the raw data are analyzed, the 20 most repeating bigrams are presented in Figure 4a. In Figure 4a, it is observed that among the most repeating terms are "sokak no" which is used to disclose people's locations, and "ni talep" and "şi kayetçi", which are used to express their demands, coupled with stop words such as "arz ederi" and "gereği ni". Figure 4b depicts the 20 most repeating bigrams terms after preparation. Sensible terms such as "hat ego" and "asphalt atilmak" are some of the most repeated bigrams after preparation. Figure 4b also depicts the classes' keywords that are likely to be connected and classes such as "hat" and "ego" that are utilized as labels in our dataset. It is easy to say that preparation has made the data more sensible.

4.2.4. Trigram Data Comparison

Figure 5a depicts the most-often-occurring trigram terms before data preparation. Word groupings that were nonsensical before preparation have become comprehensible and coherent after preparation. To see the associated terms, the trigrams give a clearer statistic. Terms such as "ni arz ederi" and "gereği ni arz" are highly repeated in trigram analysis before preparation, as shown in Figure 5a, while terms such as "hat ego otobus" and "durak hat ego" are highly repeated after data preparation, as shown in Figure 5b. Again, here, the data make more sense after the preparation.

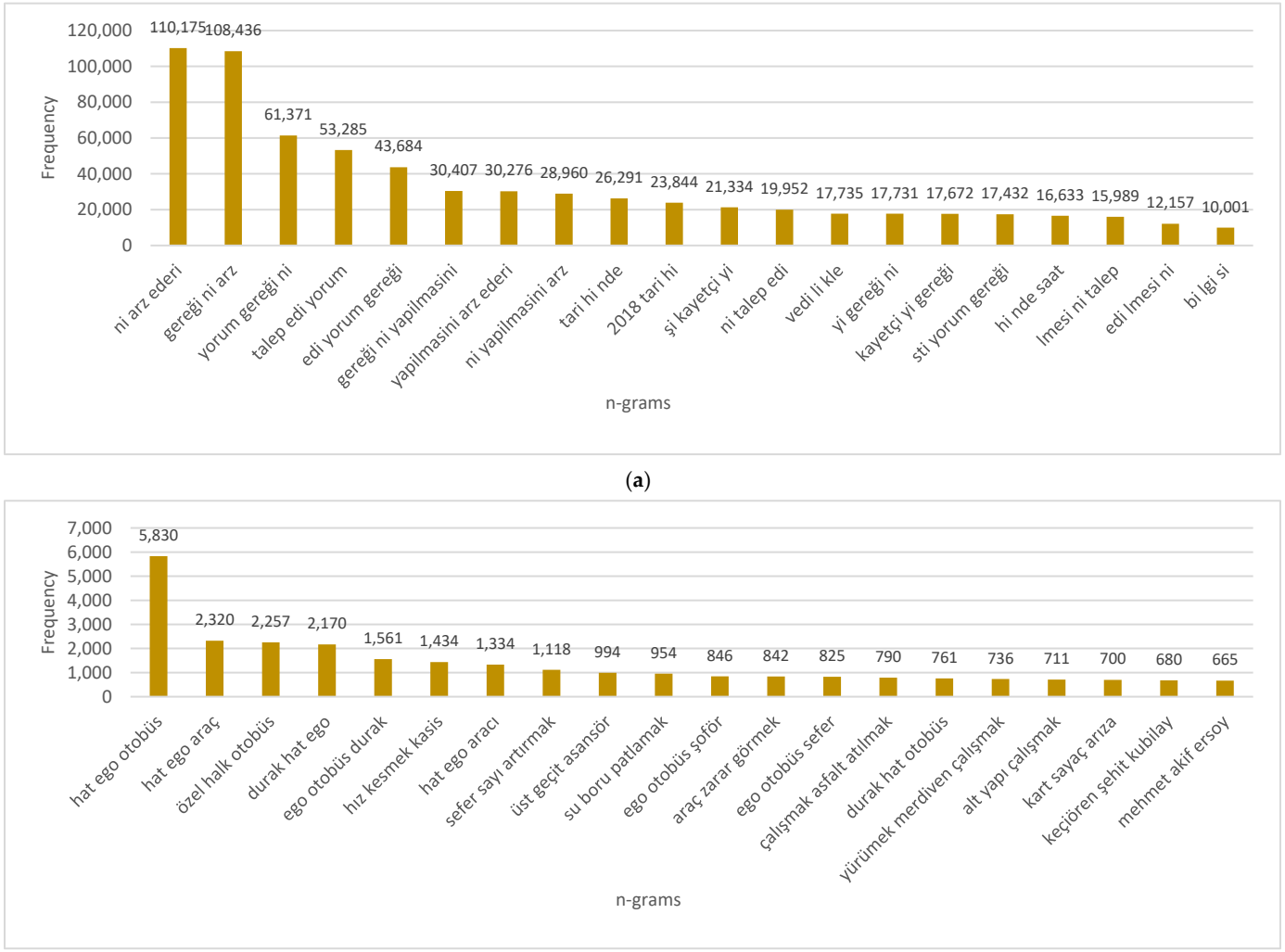
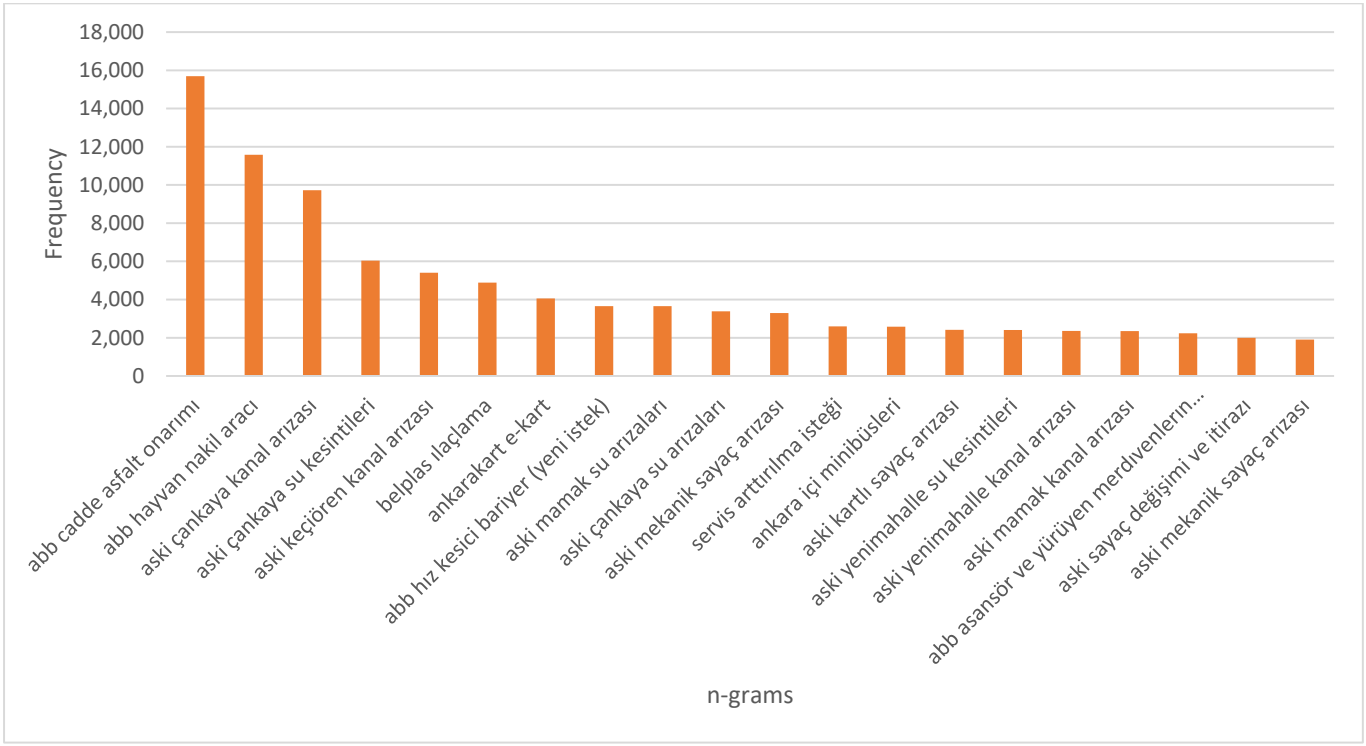


Figure 5. Most commonly occurring trigrams before and after data preparation: (**a**) most commonly occurring trigrams before preparation; (**b**) most commonly occurring trigrams after preparation.

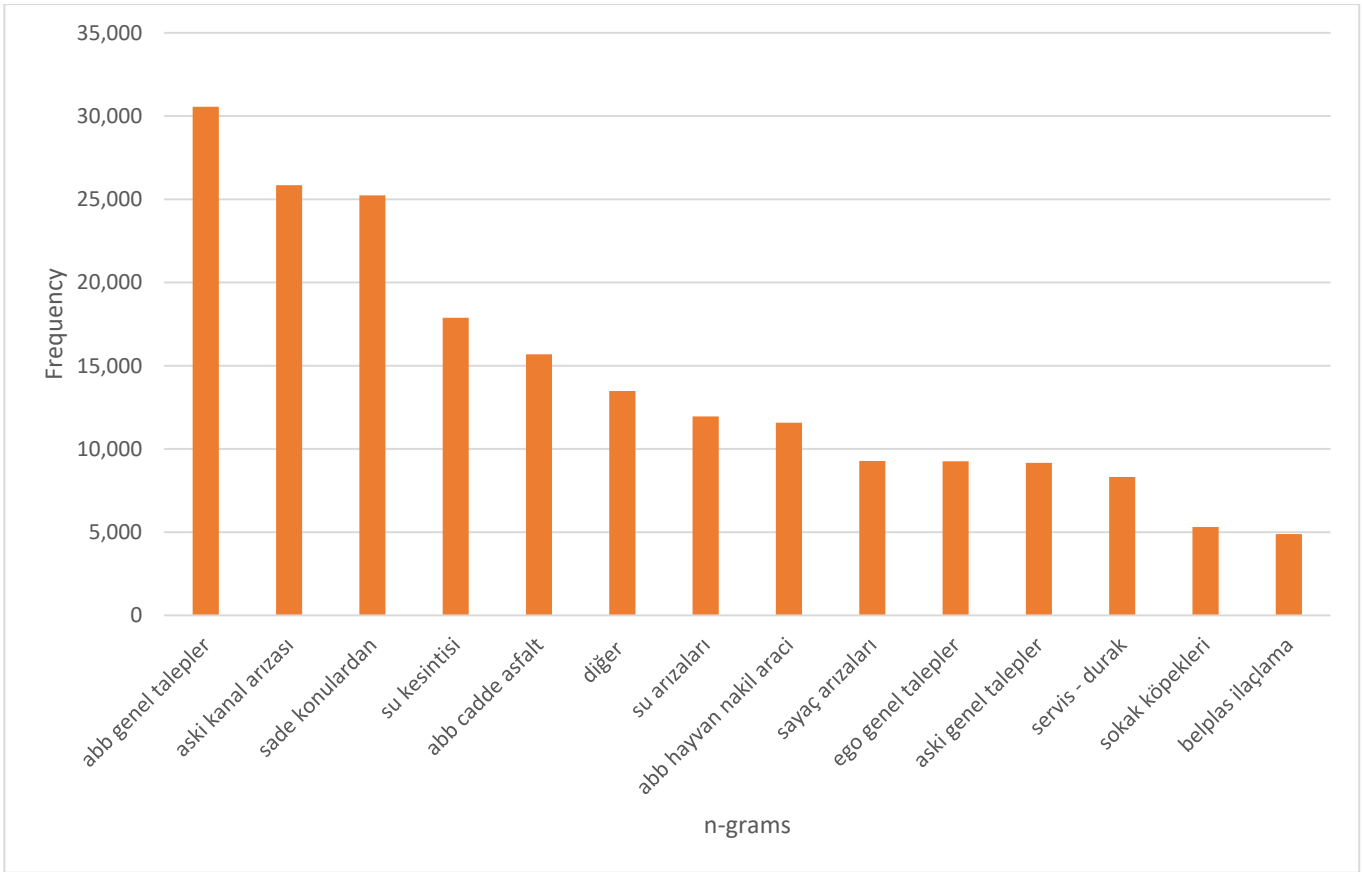
4.2.5. Data Classification

The classes in the corpus are shown to be unbalanced. This unbalanced distribution and the fact that certain classes only have one inquiry are expected to have a detrimental impact on the algorithms' performance. Figure 6a depicts the number of classes received by 20 random selection inquiries out of 1819 inquiries. The classes with fewer than 100 inquiries each were combined into one class, and classified as "others" in this study, to create more consistent data. Furthermore, several class labels appear to relate to similar classes. In Figure 6a, for example, "yenmahalle su kesntler" and "ankaya su kesntler" both relate to "su kesntler". Other classes that were comparable to this class were likewise simplified.

The distribution of the data after simplification is shown in Figure 6b. The class number was dropped from 1819 to 14. The manual reduction of the number of classes to 14 assisted us in determining the number of likely clusters. As a result, we determined that 14 clusters were necessary to execute clustering using the k-means technique. Classification methods were also applied to the new corpus, and the findings were compared.



(**a**)



(**b**)

Figure 6. Classes of inquiries before and after data preparation: (**a**) classes of inquiries before preparation; (**b**) classes of inquiries after preparation.

Figure 6 illustrates improvements made in the classes, wherein applications with similar topics were directed to the same departments. To achieve this, similar topics were grouped together using the k-means algorithm. For example, the class "ASKİ Çankaya Kanal Arızası" in Figure 6 can be formulated as "BirimAdı DistrictName Konu", where ASKİ represents the department handling canal malfunctions and is followed by the district's name and class. Through the k-means algorithm, it is simplified to "BirimAdı Konu".

4.3. Feature Extraction

The prepared data cannot be transferred straight to ML algorithms. To function, most ML algorithms require feature vectors. Therefore, the dataset must be turned into feature vectors before using ML algorithms for classification. One of the most prevalent strategies is the "Bag-of-Words" method. This method is used to calculate the total number of words in a text. Features such as word location and the order of a word are disregarded. Each word's occurrence method is represented as a feature, and the word's frequency is represented as a feature value in the bag-of-words method [43]. The bag-of-words-method-based TF-IDF was employed in this study.

Term Frequency (TF) is concerned with the number of times a term appears in a text. The recurrence of a term is significant for TF. Inverse Document Frequency (IDF) is a technique for determining which term is more essential. The significances of all the terms analyzed are considered equal when calculating TF. It is not important whether the words under consideration are stopping words or inconsequential conjunctions. Integrating TF and IDF solves this problem [23]. The relevance of the terms in the corpus texts is assessed using TF-IDF. The TF-IDF equation is shown below [14,44]. Here $tf_{i,j}$ = the total number of occurrences of i in j, df_i = the total number of documents containing i, and N = the total number of documents.

$$tfidf_{i,j} = tf_{i,j} \times log\left(\frac{N}{df_i}\right) \quad (1)$$

5. Findings

Before and after preparation, the dataset was trained using ML techniques to build models. Of the dataset, 25%, comprising 225,239 customer inquiries, was utilized as test data, with the remainder serving as training data. The RF algorithm is a type of cluster algorithm. It can be assumed to be a forest filled with DTs. When utilizing Scikit-learn with an RF classifier, the argument must specify the number of DTs that must be included. A forest with 200 DTs was employed in this investigation. The maximum depth was specified as Sckit-learn, the SVM, the NB algorithm, and LR were used to construct these models. The linear kernel was utilized in the SVM. The obtained models were evaluated using the test data, and their F1-score, recall, precision, and accuracy values were contrasted.

5.1. Performance Metrics

The prediction accuracy, loss, validation loss, and validation accuracy of ML algorithms are often used to assess them. Taking these parameters into account, we examined the output of the LSTM-trained model. The values were then chosen to illustrate whether or not the model was overfitting. A model has overfitted if the validation loss is greater than the training loss. It may be considered that an accurately fitted model has been built if both validation values are equal or extremely close to one another. Accuracy is a straightforward and widely used assessment metric. Accuracy aids in determining an algorithm's performance as a classifier. When fresh data are received, it offers the likelihood of prediction. When analyzing the performance of an ML algorithm, it is also useful to consider three additional metrics, termed precision, F1-score, and recall, in addition to its accuracy.

The equations below show the formulae for deriving F1-score, precision, accuracy, and recall. The F1-score is determined based on both precision and recall values. Accuracy, however, tells us that the number of projected classes is correct. Accuracy alone is insufficient, especially in datasets where certain classes contain a large amount of data. The

recall value indicates the rate of successfully recognizing a class. For instance, the dataset in the detection of cancer, which is a common working field, particularly in ML research, is divided into two groups. A major portion of the sample is made up of healthy persons. Typically, the system will forecast that the incoming data will come from healthy persons. Because of the uneven nature of our dataset, analyzing merely the accuracy is insufficient.

$$Accuracy = \frac{T_{positive} + T_{negative}}{T_{positive} + T_{negative} + F_{positive} + F_{negative}} \quad (2)$$

$$Precision = \frac{T_{positive}}{T_{positive} + F_{positive}} \quad (3)$$

$$Recall = \frac{T_{positive}}{T_{positive} + F_{negative}} \quad (4)$$

$$F1_score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Here, True positives are samples that were appropriately classified as positive. False positives are samples that were wrongly classified as positive. False negatives are negative samples that were wrongly classified. True negatives are samples that were appropriately classified as negative.

5.2. Model Performance Evaluation

We investigate the suggested model's performance metrics in this section.

5.2.1. Model Evaluation of Raw Data

Table 5 displays the findings obtained from the raw data. Among other algorithms, the SVM has the highest accuracy, precision, and recall values. The majority of the relevant results are returned by a high recall value. The accuracy of the RF classifier is relatively poor and does not appear to operate well with the data utilized in this study. The accuracy and F1-score values are fairly low. Even though the data are raw, the SVM and LR have more than 50% accuracy. Given the unbalanced nature of the data utilized in this study, it is necessary to examine recall and F1-score values in addition to accuracy to assess the ML algorithms' performance.

Table 5. Results before data preparation.

Algorithm	Precision	Accuracy	F1-Score	Recall
RF	9%	10%	10%	12%
NB	29%	35%	29%	29%
LR	50%	56%	52%	54%
SVM	60%	59%	64%	64%

5.2.2. Model Evaluation after Data Preparation

Table 6 displays the findings obtained after preparing corpus-simplified and balanced data. These results demonstrate that the SVM is the best ML method for our dataset. According to the findings, the SVM has a high performance with an accuracy of 78%, precision of 77%, and recall of 78%. When the unbalanced data in Table 5 is compared to the findings of Table 6, an 19% rise in accuracy, a 17% rise in precision, a 13% rise in F1-score, and a 14% rise in recall value are observed. After data preparation, all algorithms' performance and accuracy scores improve significantly. In binary classification, DTs perform better. The performance of the RF method is fairly low since the data utilized in this investigation correspond to more than two classes. When Tables 5 and 6 are compared, it is clear that

the consistency of the number of classes and the balanced data distribution significantly improve the algorithm's performance.

Table 6. Results after data preparation.

Algorithm	Precision	Accuracy	F1-Score	Recall
RF	33%	18%	7%	18%
NB	72%	70%	66%	70%
LR	76%	76%	75%	77%
SVM	77%	78%	77%	78%

5.2.3. Training Time Evaluation

The effectiveness of the algorithms utilized in this work was assessed using their training time and recall, accuracy, and precision. This is due to the fact that unclean data affect not just an algorithm's performance metrics (e.g., accuracy and precision) but also its training time [1]. The experiment's system contains an Intel® Core™ i7-6820HQ 2.70 GHz CPU and 32 GB of RAM. Regarding the provided dataset, the speed of each algorithm varied from the others. The training times for each algorithm for the raw and prepared data are displayed in Table 7 below in seconds. The table shows that all algorithms used to process the raw data had long training periods. The prepared data appears to provide shorter training. The NB scored the shortest training time. The computations, the number of classes, and the number of iterations that the algorithms must go through to finish their process of learning are all directly proportional. Due to the disparity of the data and the growing number of classes, it is evident that learning takes longer.

Table 7. Training time comparison (seconds).

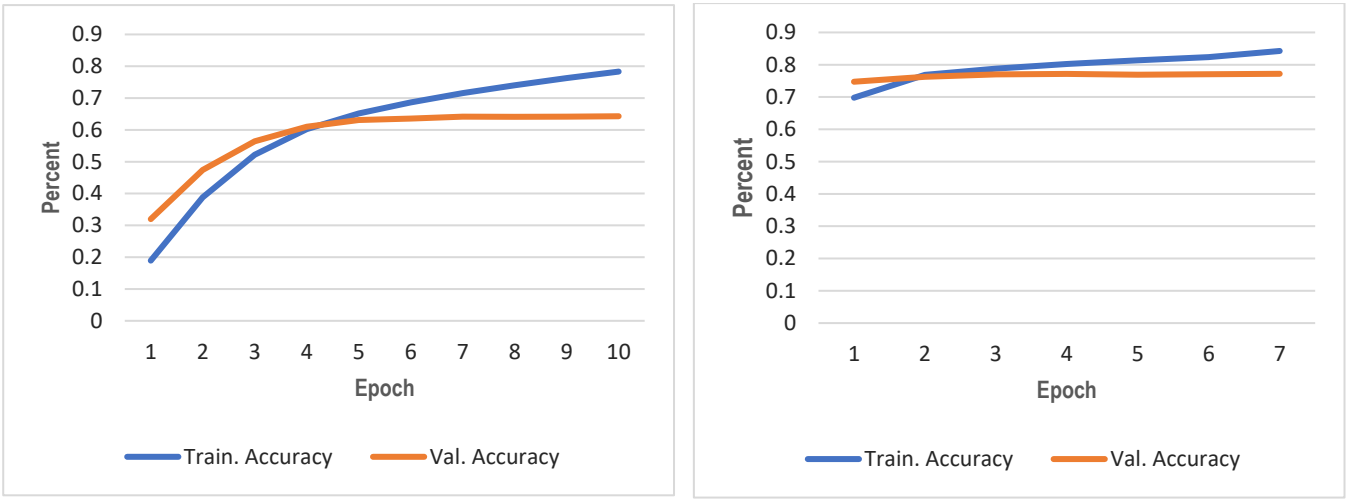
Data	RF	LR	NB	SVM	LSTM
Raw	36.727	5850.307	41.663	46,943.523	60,214.325
Prepared	7.445	39.614	0.227	8865.088	10,426.658

5.2.4. Long Term-Short Memory Insights

It can be argued that we achieved higher accuracy using the model trained using the LSTM technique than when using conventional ML algorithms. When using Keras to train the model, we provided parameters to the LSTM. To set the model's input layer, we included the embedding layer used with text data. Words are densely represented by the embedding layer. The number of inputs, input dimension, and output size are all parameters for the embedding algorithm. Since our dataset contains several classes, we employed the SoftMax function as the activation function in this study. The number of classes affected the output layer. We adjusted the loss function to categorical cross-entropy since our dataset had several classes.

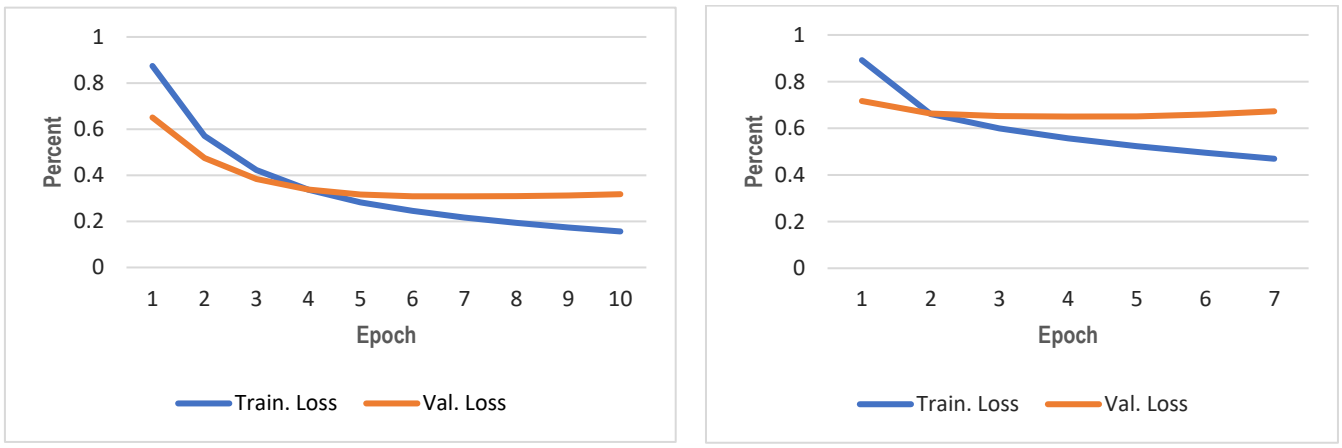
The size of the batch was provided as 64 while the size of the epoch was 10. Additionally, the 1D layer spatial dropout, which describes the likelihood of adjusting hidden layer outputs to 0, was added, and we assigned it a 0.2 rate. Instead of just deleting a single element, it deletes whole 1D feature maps. To avoid overfitting, we incorporated spatial dropout into the model [45]. The optimizer chosen was Adaptive Moment Estimation (Adam), which has the advantages of requiring little memory, performing well with big datasets, and being simple to use. The labeled data, the input data, the callback parameters, and the validation split were provided when utilizing the fit function. As the training progressed worsened, we abruptly ended it. We have 100 memory units in our LSTM layer. Finally, using the specified parameters, we examined the customer inquiries before and after data preparation.

The parameter for Keras was set to 10 epochs; however, after the 7th epoch, the training was over. Early halting, indicated by the parameter, recognized that the model had not improved and halted training. As a result, rather than after the 10th epoch, the training was finished after the 7th epoch. Figure 7 shows that the accuracy value of the LSTM, which was applied to the original data, was 78% in the most recent epoch. There was no significant overfitting issue in the model training, as seen by the decreasing validation loss and loss values and the rising values of accuracy and validation accuracy with more epochs. The model trained using the prepared data had an accuracy rating of 84%. When compared to 1.55 before preparation, the data loss was around 0.67 after data preparation, as shown in Figure 8.



(**a**) (**b**)

Figure 7. Accuracy value of the LSTM: (**a**) accuracy plot before data preparation; (**b**) accuracy plot after data preparation.



(**a**) (**b**)

Figure 8. Data loss: (**a**) loss plot before data preparation; (**b**) loss plot after data preparation.

Our results are consistent with Zulqarnain et al. [31], Karasoy and Ballı [22], and Yildiz [20]'s findings in that DL techniques, such as the LSTM, are more accurate compared to traditional ML techniques. Our results are also consistent with [26,33], while opposed to the findings of [27] in that the data preparation phase is critical in text classification.

6. Conclusions and Future Work

Text classification has had a significant influence on domains such as news categorization, the detection of spam content, and semantic analysis. The classification of Turkish text is the focus of this work since only a few studies have been conducted in this context. We utilized data obtained from letters of request that came to an institution to evaluate the proposed techniques. Five distinct classification techniques were used in this work, and the findings were thoroughly reviewed. The techniques were applied to the raw data and the preprocessed data to investigate the impacts of data preparation. The results then were compared utilizing accuracy, training time, F1-score, and recall values. Data were normalized, morphologically analyzed, processed through word stop, and simplified using the k-means technique. The data were trained using the SVM, NB, LTSM, RF, and LR techniques. The performance of the various techniques was analyzed after and before data preparation.

The LTSM was found to be the most effective technique in terms of accuracy and training time. The findings show that normalized data, as well as coherence between the categories' number and the number of training sets, are significant variables influencing the techniques' performance. Additionally, the techniques performed better once the number of categories in the dataset was reduced. The comparison also demonstrates that of all the preparation stages, the simplifying phase has the biggest influence on the results. The TF-IDF approach utilized in the feature extraction stage, on the other hand, has the biggest impact on the results. Overall, the findings of this study achieved 84% performance accuracy, which is considerably higher than those of previously proposed text classification solutions. Furthermore, the text categorization approach utilized in this work applies to data in languages other than Turkish. However, in the morphological analysis stage of a comparable study, one must employ a strategy appropriate to their languages.

In NLP jobs, it is critical to analyze data for future research and analyses. The fact that the data in these jobs are based on natural languages (from humans) may lead to excess data filth. Furthermore, data utilized in algorithms for supervised learning must be classified repeatedly for a trained model to produce accurate findings. More efficient outcomes can be obtained by training the model using a classifier more consistently. That is, once the dataset is expanded with fresh data, an LSTM technique with higher accuracy may be provided. With data cleansing, it is also feasible to decrease the time required for the preparation.

Author Contributions: Conceptualization, A.E.E. and A.E.T.; methodology, Y.I.A.; software, A.E.E.; validation, Y.I.A. and A.E.T.; writing—original draft preparation, A.E.E.; writing—review and editing, Y.I.A.; supervision, A.E.T. All authors contributed to the study conception and design. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions on privacy or ethical.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ajitha, P.; Sivasangari, A.; Immanuel Rajkumar, R.; Poonguzhali, S. Design of text sentiment analysis tool using feature extraction based on fusing machine learning algorithms. *J. Intell. Fuzzy Syst.* **2021**, *40*, 6375–6383. [CrossRef]
2. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep learning-based text classification: A comprehensive review. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–40. [CrossRef]
3. Srinivasan, S.; Ravi, V.; Alazab, M.; Ketha, S.; Al-Zoubi, A.M.; Kotti Padannayil, S. Spam emails detection based on distributed word embedding with deep learning. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Studies in Computational Intelligence*; Maleh, Y., Shojafar, M., Alazab, M., Baddi, Y., Eds.; Springer: Cham, Germany, 2021; Volume 919, pp. 161–189.

4. Akhter, M.P.; Jiangbin, Z.; Naqvi, I.R.; Abdelmajeed, M.; Fayyaz, M. Exploring deep learning approaches for Urdu text classification in product manufacturing. *Enterp. Inf. Syst.* **2022**, *16*, 223–248. [CrossRef]
5. Sarker, I.H. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.* **2021**, *2*, 160. [CrossRef] [PubMed]
6. Mohammed, A.; Kora, R. An effective ensemble deep learning framework for text classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 8825–8837. [CrossRef]
7. Qasim, R.; Bangyal, W.H.; Alqarni, M.A.; Ali Almazroi, A. A fine-tuned BERT-based transfer learning approach for text classification. *J. Healthc. Eng.* **2022**, *2022*, 3498123. [CrossRef]
8. Thirumoorthy, K.; Muneeswaran, K. Feature selection for text classification using machine learning approaches. *Natl. Acad. Sci. Lett.* **2022**, *45*, 51–56. [CrossRef]
9. Luo, X. Efficient english text classification using selected machine learning techniques. *Alex. Eng. J.* **2021**, *60*, 3401–3409. [CrossRef]
10. Altınel, B.; Ganiz, M.C. Semantic text classification: A survey of past and recent advances. *Inf. Process. Manag.* **2018**, *54*, 1129–1153. [CrossRef]
11. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
12. Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–41. [CrossRef]
13. Hartmann, J.; Huppertz, J.; Schamp, C.; Heitmann, M. Comparing automated text classification methods. *Int. J. Res. Mark.* **2019**, *36*, 20–38. [CrossRef]
14. Shah, K.; Patel, H.; Sanghvi, D.; Shah, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augment. Hum. Res.* **2020**, *5*, 12. [CrossRef]
15. El Rifai, H.; Al Qadi, L.; Elnagar, A. Arabic text classification: The need for multi-labeling systems. *Neural Comput. Appl.* **2022**, *34*, 1135–1159. [CrossRef] [PubMed]
16. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
17. Dai, Y.; Guo, W.; Chen, X.; Zhang, Z. Relation classification via LSTMs based on sequence and tree structure. *IEEE Access* **2018**, *6*, 64927–64937. [CrossRef]
18. Yuvaraj, N.; Chang, V.; Gobinathan, B.; Pinagapani, A.; Kannan, S.; Dhiman, G.; Rajan, A.R. Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. *Comput. Electr. Eng.* **2021**, *92*, 107186. [CrossRef]
19. Yadav, B.P.; Ghate, S.; Harshavardhan, A.; Jhansi, G.; Kumar, K.S.; Sudarshan, E. Text categorization performance examination using machine learning algorithms. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Warangal, India, 9–10 October 2020; IOP Publishing: Warangal, India, 2020; p. 022044.
20. Yildiz, B. Efficient text classification with deep learning on imbalanced data improved with better distribution. *Turk. J. Sci. Technol.* **2022**, *17*, 89–98. [CrossRef]
21. Köksal, Ö.; Yılmaz, E.H. Improving automated Turkish text classification with learning-based algorithms. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6874. [CrossRef]
22. Karasoy, O.; Ballı, S. Spam SMS detection for Turkish language with deep text analysis and deep learning methods. *Arab. J. Sci. Eng.* **2022**, *47*, 9361–9377. [CrossRef]
23. Bozyigit, F.; Dogan, O.; Kilinc, D. Categorization of customer complaints in food industry using machine learning approaches. *J. Intell. Syst. Theory Appl.* **2022**, *5*, 85–91.
24. Amasyalı, M.F.; Diri, B. Automatic Turkish text categorization in terms of author, genre and gender. In *Natural Language Processing and Information Systems. NLDB 2006. Lecture Notes in Computer Science*; Kop, C., Fliedl, G., Mayr, H.C., Métais, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3999, pp. 221–226.
25. Güran, A.; Akyokuş, S.; Bayazıt, N.G.; Gürbüz, M.Z. Turkish text categorization using n-gram words. In Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2009), Trabzon, Turkey, 29 June–1 July 2009; IEEE: Trabzon, Turkey, 2009; pp. 369–373.
26. Uysal, A.K.; Gunal, S. The impact of preprocessing on text classification. *Inf. Process. Manag.* **2014**, *50*, 104–112. [CrossRef]
27. Yıldırım, S.; Yıldız, T. A comparative analysis of text classification for Turkish language. *Pamukkale Univ. J. Eng. Sci.* **2018**, *24*, 879–886. [CrossRef]
28. Kuyumcu, B.; Aksakalli, C.; Delil, S. An automated new approach in fast text classification (fastText): A case study for Turkish text classification without pre-processing. In Proceedings of the 3rd International Conference on Natural Language Processing and Information Retrieval, ACM, Tokushima, Japan, 28–30 June 2019; pp. 1–4.
29. Çoban, Ö.; Özel, S.A.; İnan, A. Deep learning-based sentiment analysis of Facebook data: The case of Turkish users. *Comput. J.* **2021**, *64*, 473–499. [CrossRef]
30. Dogru, H.B.; Tilki, S.; Jamil, A.; Hameed, A.A. Deep learning-based classification of news texts using doc2vec model. In Proceedings of the 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; IEEE: Riyadh, Saudi Arabia, 2021; pp. 91–96.

31. Zulqarnain, M.; Alsaedi, A.K.Z.; Ghazali, R.; Ghouse, M.G.; Sharif, W.; Husaini, N.A. A comparative analysis on question classification task based on deep learning approaches. *PeerJ Comput. Sci.* **2021**, *7*, e570. [CrossRef] [PubMed]
32. Bektaş, J. Detection of economy-related Turkish tweets based on machine learning approaches. In *Data Mining Approaches for Big Data and Sentiment Analysis in Social Media*; El-Latif, A.A.A., Ed.; IGI Global: Hershey, PA, USA, 2022; pp. 171–195.
33. Eminagaoglu, M. A new similarity measure for vector space models in text classification and information retrieval. *J. Inf. Sci.* **2022**, *48*, 463–476. [CrossRef]
34. Erkaya, A.E. *Text Classification based on Organizational Data Using Machine Learning*; Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü: Keçiören/Ankara, Türkiye, 2019.
35. Akın, A.A.; Akın, M.D. Zemberek, an open source NLP framework for Turkic languages. *Structure* **2007**, *10*, 1–5.
36. Kayabaş, A.; Schmid, H.; Topcu, A.E.; Kiliç, Ö. TRMOR: A finite-state-based morphological analyzer for Turkish. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 3837–3851. [CrossRef]
37. Pandas. User Guide. NumFOCUS, Inc. Hosted by OVHcloud. 2022. Available online: https://pandas.pydata.org/docs/user_guide/index.html (accessed on 25 July 2022).
38. Matplotlib. Matplotlib: Visualization with Python. 2022. Available online: https://matplotlib.org (accessed on 27 July 2022).
39. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
40. Keras. Developer Guides. 2019. Available online: https://keras.io/guides/ (accessed on 29 July 2022).
41. Akın, A.A. zemberek-nlp. 2021. Available online: https://github.com/ahmetaa/zemberek-nlp (accessed on 15 August 2022).
42. Jaradat, A.; Safieddine, F.; Deraman, A.; Ali, O.; Al-Ahmad, A.; Alzoubi, Y.I. A probabilistic data fusion modeling approach for extracting true values from uncertain and conflicting attributes. *Big Data Cogn. Comput.* **2022**, *6*, 114. [CrossRef]
43. Zhang, Z.-H.; Min, F.; Chen, G.-S.; Shen, S.-P.; Wen, Z.-C.; Zhou, X.-B. Tri-partition state alphabet-based sequential pattern for multivariate time series. *Cogn. Comput.* **2022**, *14*, 1881–1899. [CrossRef]
44. Hossain, T.; Mauni, H.Z.; Rab, R. Reducing the effect of imbalance in text classification using SVD and GloVe with ensemble and deep learning. *Comput. Inform.* **2022**, *41*, 98–115. [CrossRef]
45. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.