Middle East Journal of Science

https://dergipark.org.tr/mejs

MEJS

e-ISSN:2618-6136

Research Article

ENHANCING MULTI-CLASS TEXT CLASSIFICATION WITH APRIORI-BASED FEATURE SELECTION

Maide Feyza ER[1]* ***Turgay Tugay BILGIN**[2]*

[1] Bandirma Onyedi Eylul University, Faculty of Engineering and Natural Sciences, Department of Software Engineering, Balıkesir, Türkiye
[2]Bursa Technical University, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Bursa, Türkiye
* Corresponding author; mer@bandirma.edu.tr

Abstract: *In the field of Natural Language Processing, selecting the right features is crucial for reducing unnecessary model complexity, speeding up training, and improving the ability to generalize. However, the multi-class text classification problem makes it challenging for models to generalize well, which complicates feature selection. This paper investigates how feature selection impacts model performance for multi-class text classification, using a dataset of projects completed by TÜBİTAK TEYDEB between 2009 and 2022. The study employs LSTM, a deep learning method, to classify the projects into nine different industries based on various attributes. The paper proposes a new feature selection approach based on the Apriori algorithm, which reduces the number of attribute combinations considered and makes model training more efficient. Model performance is evaluated using metrics like accuracy, loss, validation scores, and test scores. The key findings are that feature selection significantly affects model performance, and different feature sets have varying impacts on performance.*

Keywords*: Feature Selection, LSTM, NLP, Text Classification*

Received: April 29, 2024 *Accepted: June 25, 2024*

1. Introduction

With the rapid development of the information age, advances in text mining and natural language processing have increased the capacity to extract meaning from large datasets. In this context, text classification has emerged as an important component of knowledge extraction. Multi-class text classification aims to classify text documents into multiple categories, and research in this area has focused on feature selection as a particularly important focus.

Feature selection is a critical factor affecting the performance of text classification models. Correct feature selection can lead to both computational savings and an improvement in the overall performance of the model [1]. This paper aims to delve deeper into the impact of feature selection on model performance in multiclass text classification. The studied effects have the potential to make the information extraction process more effective and improve efficiency in text mining applications.

Thirumoorthy et al. presented a feature selection technique based on the frequency distribution measure to address the high-dimensional feature space problem of text classifier accuracy reduction in the setting of a very high-dimensional feature space. In their study, the authors used SVM and Naive Bayes classifiers on two benchmark datasets to assess the efficacy of the suggested feature selection technique. They reported that, in terms of classification accuracy, the recommended feature selection method performed better than alternative feature selection strategies [2].

Amazal et al. on the other hand, proposed a distributed feature selection approach for large-scale multi-label textual big data using the weighted chi-square method implemented in the Hadoop framework. The suggested approach, which transforms multi-label data into single-label data, assigns weights to features based on the category term frequency and calculates the chi-square for each feature according to its weight. Experimental results have demonstrated that the proposed method is efficient, robust, and scalable when compared to state-of-the-art techniques [3].

Naik et al. highlighted the significance of text preprocessing, which entails organizing and cleaning data, in their work addressing the difficulties of processing and interpreting massive volumes of unstructured textual data. They emphasized how important it is to use Natural Language Processing (NLP) techniques to prepare data for analysis in order to extract useful information from text [4]. These techniques include language identification, tokenization, filtering, lemmatization, and stemming.

Dowlagar and Mamidi proposed a hybrid feature selection method that combines various filter-based feature selection techniques with the fastText classifier to obtain the necessary features for text classification. They observed a reduction in training time and a slight increase in accuracy in some datasets when feature selection methods were employed in conjunction with neural networks [5].

Hussain and colleagues proposed a novel feature ranking score called the Differential Mutual Information (DMI) score and an avant-garde technique called Non-Redundant Feature Selection (NRFS) in their work to solve the shortcomings of conventional feature selection methods in text data. In comparison to previous state-of-the-art methods, the suggested method was shown to produce superior micro-F1 classification scores and to exhibit more resilience against label noise, especially in situations when there were few picked features [6].

Belkarkor and colleagues’s research centered on the difficulty of handling large amounts of data in text processing and the significance of feature selection in machine learning algorithms for text categorization. With the use of three reference document collections and the NB classifier, the study assessed the effectiveness of Genetic Algorithm in feature selection by contrasting it with alternative filtering techniques. According to the study, Genetic Algorithm fared better in text categorization than other filtering techniques in terms of efficiency and accuracy [7]. It was noted that the best feature selection strategy differed for every dataset in Zheng's paper, which compared various feature selection techniques. Nonetheless, some techniques regularly yielded valuable data for class categorization, and chi-square was widely acknowledged as the best technique [8].

In order to improve the efficacy of feature selection in text mining and machine learning, Tang et al. presented a feature selection technique that uses two to five-way interactions to account for high-order feature interactions. By breaking down the mutual information-based feature selection problem into lower-order interactions, the strategy depends on an effective measurement for predicting interaction terms. They claimed that by taking into account high-level feature interactions, their suggested approach performed better when it came to feature selection for text categorization tests [9].

By analyzing various feature selection techniques and comprehending the variances in the performance of multi-class text classification models, this study aims to give academics important insights into overcoming difficulties presented by multi-featured datasets. This paper will examine pertinent research in natural language processing and text mining, highlighting the critical role feature selection plays from a strategic standpoint. Next, the emphasis will be on the findings of research investigations carried out to provide insight into how feature selection affects text classification model performance.

In this context, an overview of the data set is presented in the Materials and Methods section of the article. Following, text preprocessing, feature selection algorithm and model training are explained respectively. The findings are discussed in the Results and Discussion section and finally the Conclusion is given in the last section.

2. Materials and Methods

This section provides a detailed explanation of the working steps. It consists of four main headings in total: Data Set Overview, Text Preprocessing, Feature Selection Algorithm, and Model Training. The method proposed in the study is presented in Figure 1.

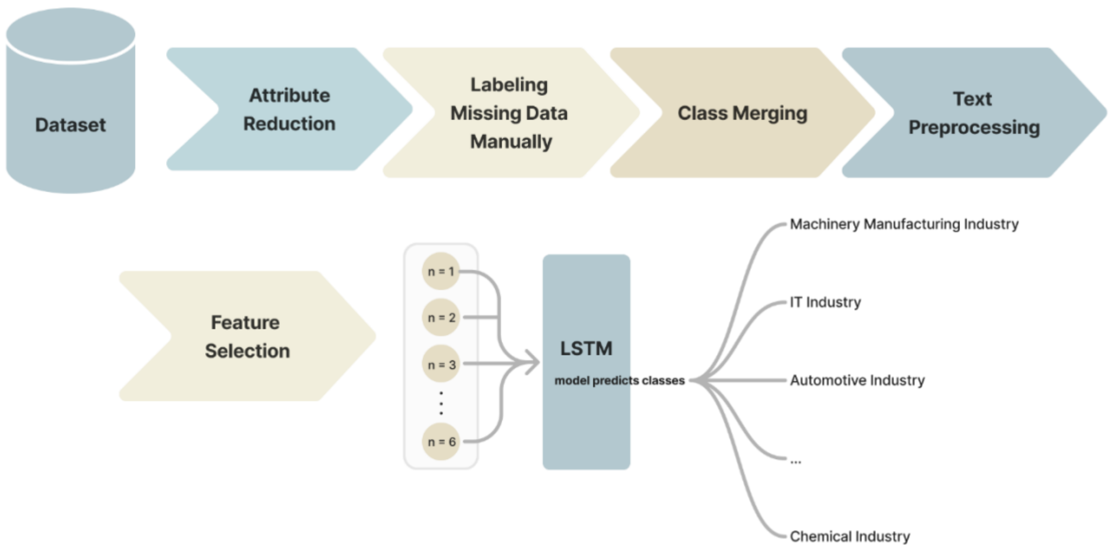

Figure 1. The proposed method in the study.

2.1. Dataset Overview

Initially, we collected the data via an RPA Tool from the TEYDEB Project Evaluation and Monitoring System portal of the TUBITAK website. The compilation comprises information about TUBITAK TEYDEB projects completed in the years 2009 through 2022. Researchers may access the details of completed projects on the TUBITAK TEYDEB website [10]. There are 1672 records with 11 attributes in the collection. Table 1 enumerates each attribute in the dataset and the number of items that correspond to that attribute.

Table 1. The dataset's attributes and the quantity of entries for each attribute.

Attributes	Number of Entries
Project Name	1672
Organization Name	1672
Organization Province	1672
Project Year	1672
Keywords	1406
Project Start-End Date	1672
Scientific Technological Activity Area	1656
Industry that R&D Activities are Conducted	1127
Industry that Project Outputs are Utilized	1038
Project Summary	671
Project Objectives	1194
Technical Specifications of Project Outputs	1449

It is evident from Table 1 that certain features in the dataset will not help to improve the classification success rate. The “Organization Name”, “Organization Province”, “Project Year”, and “Project Start-End Date” are examples of these features. They have been removed from the dataset.

Table 1 also shows that certain attributes in the dataset have missing data. There are two main methods for handling missing data. The first and easiest option is to remove missing data; the second method uses random, statistical, or k-nearest neighbor algorithms to fill in the missing data. However, it should be noted that these methods may have adverse effects on the dataset and are not universally applicable to every situation. Deleting missing data can lead to further reduction in small datasets, causing data loss. Additionally, filling in missing data is generally applicable only to numerical values.

The "Industry that Project Outputs will be used" attribute will serve as the basis for classification, hence any missing data in this attribute has to be manually labeled because removing it would result in a significant loss of data. The “2.3 Feature Selection” section provides an explanation of the method used to select the solution that is deemed most appropriate for the dataset's characteristics. Manually filling in the missing values for the other attributes is not feasible. After manually filling, the number of classes for the attribute's classification and the total number of data in these classes were analyzed. Table 2 provides the number of projects grouped by the Industries.

Table 2. Number of projects grouped by the industries

The industry that Project Outputs will be Utilized	Number of Data
Machinery Manufacturing Industry	354
IT Industry	217
Automotive Industry	175
Electrical & Electronics Industry	130
Biomedical Industry	112
Energy Industry	103
Food Industry	83
Material Industry	83
Textile Industry	69
Chemical Industry	66
Defense Industry	43
White Goods Industry	37
Metallurgical Industry	35
Pharmaceutical Industry	30
Telecommunications Industry	30
Agriculture Industry	29
Environmental Technologies Industry	21
Ship and Maritime Industry	13
Aerospace Industry	11
Mining Industry	10
Livestock Industry	10
Aquaculture Industry	9
Ceramic/Earth Products	2

There are a total of 23 classes in the dataset, as Table 2 shows, and the distribution of the number of entities is not balanced. There are many classes available for classification despite the small size of the dataset. This is an important issue that can make the classifier less effective. The merging of closely related classes has been considered as an improvement technique, even if it would not be possible to fully solve this issue. As a result, there would be fewer classes and more entities in each class.

Data from the Agriculture and Livestock Industry are combined with the Food Industry. In addition, the Pharmaceuticals-Pharmaceuticals Industry was merged with the Chemicals Industry. The Metallurgy Industry was merged with the Materials Industry. The Telecommunications Industry was merged with the Information Technology Industry. Lastly, the Household Appliances Industry with the Electrical and Electronics Industry merged. Classes with less than 90 entities have been removed from the dataset after these merges. As a result, the final situation regarding industries and number of projects is shown in Table 3.

Table 3. The dataset's attributes and the quantity of entries for each attribute.

Industries that Project Outputs will be Utilized	**Number of Data**
Machinery Manufacturing Industry	354
IT Industry	247
Automotive Industry	175
Electrical & Electronics Industry	167
Food Industry	122
Material Industry	118
Biomedical Industry	112
Energy Industry	103
Chemical Industry	96

After merging, classes and their respective data densities are displayed in Figure 2. The distribution of the classes is found to be more uniform than it was in the original data set. It is clear that the classifiers will perform better in this way.

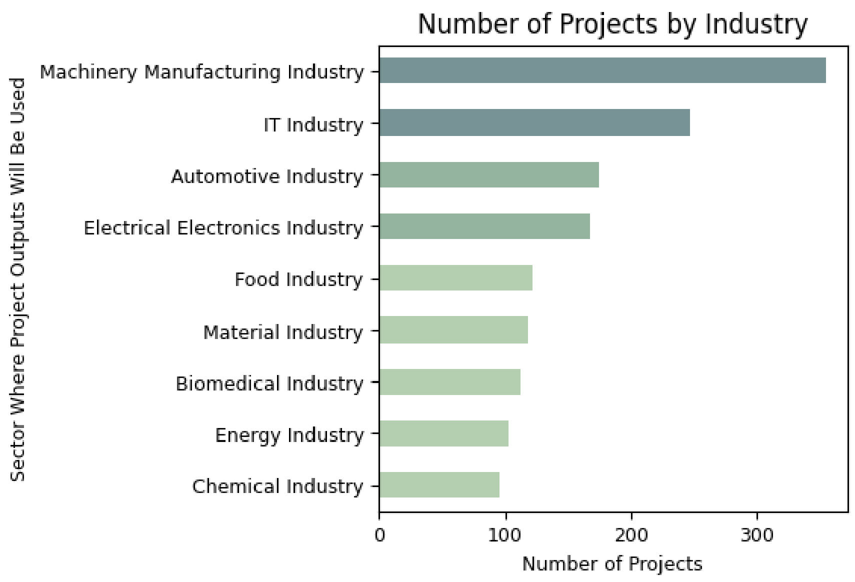

Figure 2. Barchart shows class labels and the data density of classes after merging.

2.2. Text Preprocessing

Text preprocessing, also known as the process of preparing raw text for natural language processing, aims to extract valuable information from the text by removing attributes that are irrelevant to the analysis. In this context, text preprocessing is critical in optimizing text for analysis by removing non-essential attributes from the dataset.

Before the dataset can be classified, we should remove user-generated textual noise and overcome difficulties specific to the Turkish language. Since all texts in the dataset will be converted into numeric values before classification, making all texts with the same meaning identical is the main goal of text preprocessing. In this context, in the preprocessing stage, Python's 're' module was utilized to check for certain patterns in the text and replace them with the desired patterns. This module enables checking whether a string of characters matches a specific regular expression. A regular expression specifies a pattern that matches a set of characters in itself [11].

Regular Expressions in Python, provided through the re-module, constitute a specialized parser. This parser allows for the establishment of rules to match specific potential word sequences. These rules can be utilized to check whether a string of characters matches a desired pattern or if there is a match for a specific pattern. Additionally, regular expressions can be used to modify a string or split it in various ways [11].

The first stage of text preprocessing involves converting all letters to lowercase. The conventional method of converting to lowercase proves inadequate in transforming Turkish characters not found in the Latin alphabet. For instance, the letter "I" is converted to lowercase as "i". To overcome this issue, specific regex rules have been defined for all Turkish-specific characters. After converting to lowercase, the text preprocessing process is completed by removing all characters and numbers except alphanumeric characters.

2.3. Feature Selection Algorithm

After removing redundant attributes from the dataset and manually filling in missing data in the "Industry that Project Outputs will be used" attribute, the final version of the dataset is provided in Table 4.

Table 4. Total number of entries for each attribute after removing redundant attributes.

Attributes	**Number of Entries**
Project name	1672
Keywords	1406
Scientific Technological Field of Activity	1656
Industry that R&D Studies will be carried out	1127
Industry that Project Outputs will be used	1672
Summary of the Project	671
The goal of the project	1194
Technical Specifications of Project Outputs	1449

As seen from Table 4, missing data still exists in all attributes except "Project Name" and "Industry that Project Outputs will be used". A classification model can be trained with the single attribute: "Project Name". However, using a single attribute will significantly decrease the model performance. Therefore, in this study, all attributes are used in various combinations while training, to investigate which attribute has a positive impact on the classifier performance. All of the attribute combinations include the "Project Name" attribute. In this context, all subsets of the attribute combinations containing “Project Name” were employed to train the model. With this approach, we intended to reduce the effect of missing data problems on model performance and to observe the effect of various attribute combinations on performance.

In the first attempt of model training, the “Project Name” attribute will be used. In the second attempt, all two element combinations containing the “Project Name” attribute will be used to train the model. In the next attempt, all three-element combinations containing the pair that provides the highest model performance among the two-element subsets will be used for model training. The same cycle will continue until there is only one subset containing all attributes.

Our proposed method for the feature selection is given in Figure 3. The approach is similar to the Apriori algorithm used for "association analysis". When the number of features is n=1, the model will be trained only with the “Project Name” feature. When n=2, all two-element combinations containing the “Project Name” attribute will be used. The subset with the highest performance from the two-element combinations will be selected, and all three-element subsets containing these two selected features will be used. In this way, all combinations including all elements will be subjected to model training.

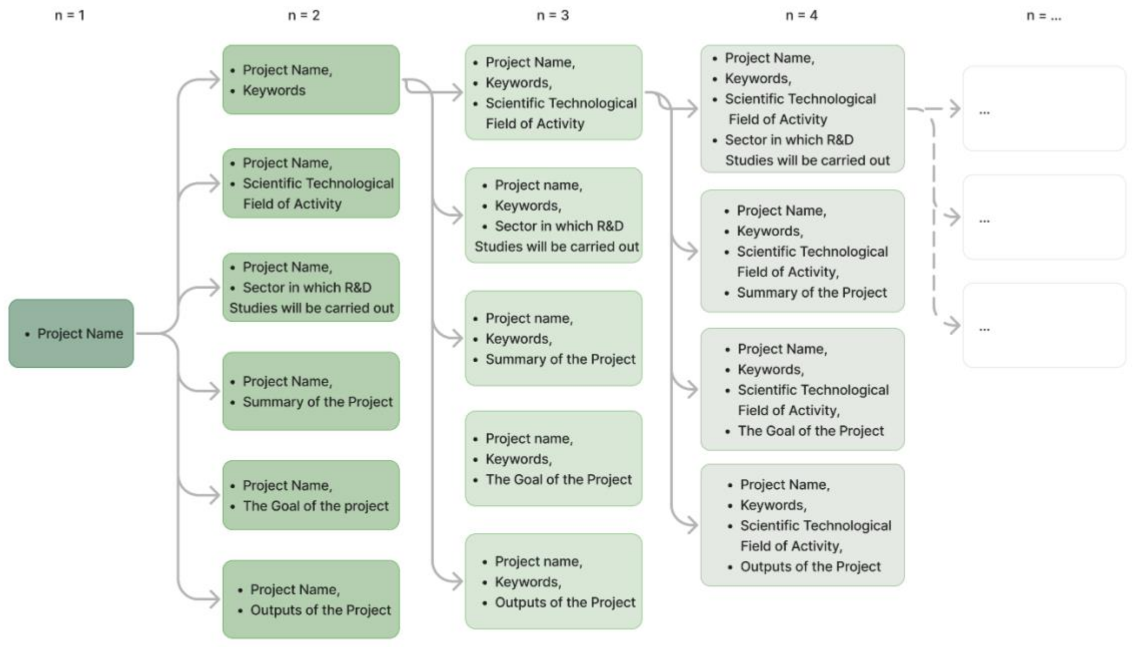
n = 1
n = 2
n = 3
n = 4
n = ...
• Project Name
• Project Name, • Keywords
• Project Name, • Scientific Technological Field of Activity
• Project Name, • Sector in which R&D Studies will be carried out
• Project Name, • Summary of the Project
• Project Name, • The Goal of the project
• Project Name, • Outputs of the Project
• Project Name, • Keywords, • Scientific Technological Field of Activity
• Project name, • Keywords, • Sector in which R&D Studies will be carried out
• Project name, • Keywords, • Summary of the Project
• Project name, • Keywords, • The Goal of the Project
• Project name, • Keywords, • Outputs of the Project
• Project Name, • Keywords, • Scientific Technological Field of Activity • Sector in which R&D Studies will be carried out
• Project Name, • Keywords, • Scientific Technological Field of Activity, • Summary of the Project
• Project Name, • Keywords, • Scientific Technological Field of Activity, • The Goal of the Project
• Project Name, • Keywords, • Scientific Technological Field of Activity, • Outputs of the Project
...
...
...

Figure 3. A representative illustration of feature selection based on the Apriori Algorithm. For illustrative purposes figure is drawn for “Project Name” and “Keywords” attributes.

To adapt the Apriori algorithm for feature selection, our proposed algorithm steps are given below:

1. **Initialization**: Start with individual features as item sets.
2. **Iteration**:
   - Generate candidate feature sets by combining existing features.
   - Evaluate the classification model accuracy using each candidate feature sets.
   - Prune candidate feature sets that do not improve classification accuracy above a predefined threshold.
3. **Termination**: Repeat the iteration until no more candidate feature sets can be generated or until the desired number of features is reached.

The key difference from traditional Apriori is in the evaluation step, where instead of measuring the frequency of item sets, we measure the classification accuracy of feature sets. The pruning step remains similar, where feature sets that do not meet the predefined accuracy threshold are discarded.

Considering only feature sets that meet the predefined accuracy threshold will significantly reduce the number of calculations required to generate new subsets. The total number of subsets for a set of N elements can be calculated using the formula for the power set. Let N be the number of elements in the set. The total number of subsets S(N) for a set of N elements is given by Eq. (1).

$$S(N) = 2^N \tag{1}$$

Since the total number of features in the dataset is 7, the total number of subsets is 128. This implies that 128 combinations of feature sets should be used for model training. On the other hand, in our proposed Apriori-like method, new subsets are generated only from the top-performing subsets. Thus, the number of trials for model training is significantly reduced.

According to the suggested feature selection approach, there are 5 three-element subsets when there are 6 two-element subsets that contain the "Project Name". The total number of viable combinations reduces by 1 as the subset's element count rises. In this scenario, it is possible to compute the total number of subgroups using Equation (2).

$$\frac{N \times (N+1)}{2} \tag{2}$$

There are 21 subsets with "Project Name" with more than one element. Only the subset with the attribute "Project Name" will be used for model training among the subsets containing a single element. As a result, a total of 22 distinct attribute sets will be used to train the model. This reduces the number of model trials from 128 to 22, yielding faster and more efficient results.

2.4. Model Training

In this study, Long Short-Term Memory (LSTM) is used for model training. LSTM was introduced as a solution to the vanishing or exploding gradient problems that arise in Recurrent Neural Networks (RNN), particularly when learning long-term dependencies [12]. Therefore, LSTMs represent a specialized type of RNN capable of learning long-term dependencies and retaining information over extended periods by default [13].

LSTM utilizes short-term and long-term memory cells to determine the importance of data. If the data is deemed important, it is stored in memory and fed back into the network; otherwise, the data is forgotten to clear the memory. LSTM has a chain structure consisting of identical cells that repeat each other [13]. The chain structure of LSTM, consisting of a series of recurring connected cells, is given in Figure 4. Each cell is designed in a special way to store and transmit information.

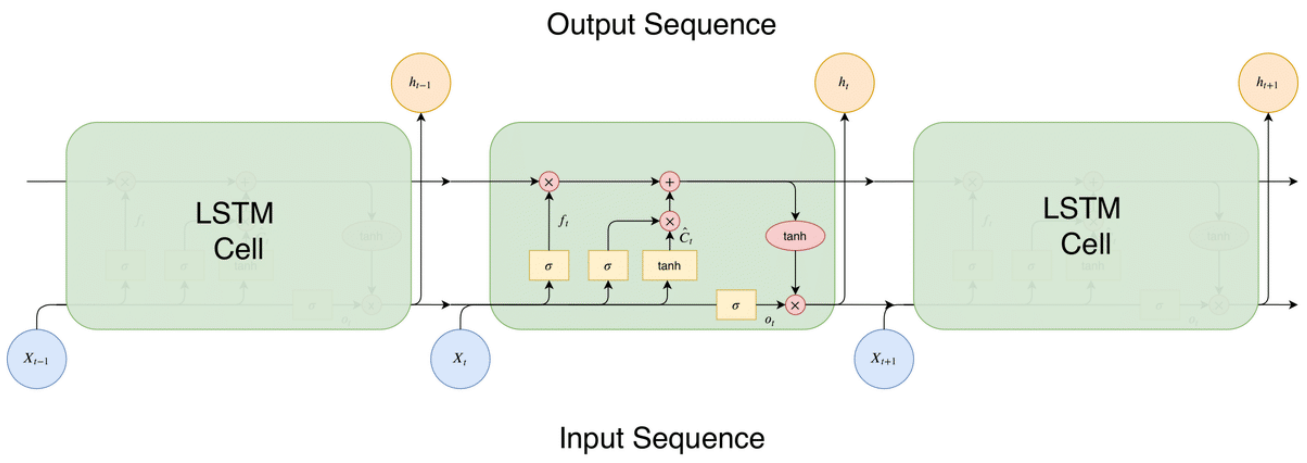

Figure 4. Chain structure of LSTM network.

The internal architecture of an LSTM cell is given in Figure 5. The basic components of an LSTM cell consist of the Cell State and three main gates: Input Gate, Output Gate and Forget Gate. While the Input Gate i_t decides when to read the data, the Forget Gate f_t is used to reset the contents of the cell. On the other hand, Output Gate o_t transfers the output of the cell to the next cell. Lastly, the Cell State, which is the main component of the data flow, ensures that the data flows forward without changing. This behavior of the Cell State reduces the vanishing gradient problem and provides a better understanding of long-time dependencies [13].

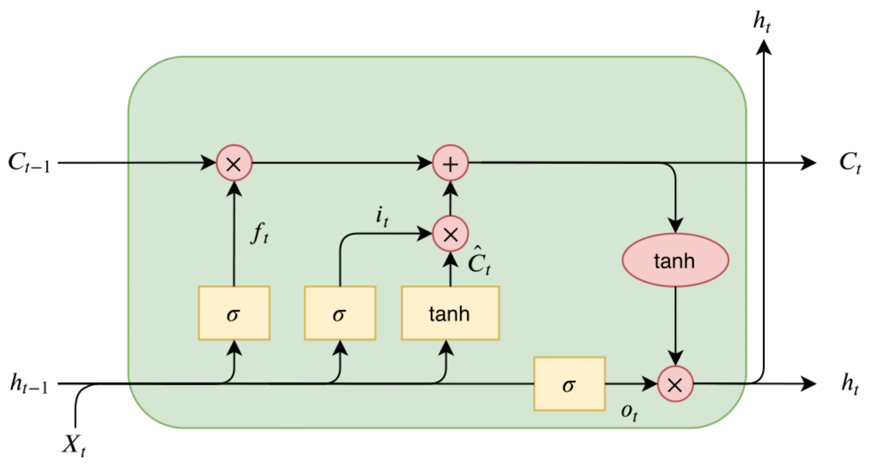

Figure 5. Cell architecture of LSTM.

As seen in Figure 5, a cell receives three inputs: X_t, (h_{t-1}) and (C_{t-1}). X_t is the input that feeds the cell at the current time t, (h_{t-1}), which is called the hidden state, is the output of the previous cell and (C_{t-1}) is the cell state of the previous cell. Subsequently, it modifies the previous memory cell (C_{t-1}) to produce two new outputs, (h_t) and (C_t). C_t is the newly updated memory, known as the cell state, while h_t is the output of the current LSTM cell, known as the hidden state. At each time step, these two states, (C_t) and (h_t) are transferred from the current cell to the next cell. Likewise, the previous cell state (C_{t-1}) and the previous hidden state (h_{t-1}) were transferred from the previous cell to the current cell at time step *t-1* [13].

The LSTM cell receives inputs, X_t and (h_{t-1}), and feeds them to three LSTM gates. These inputs pass through the sigmoid, which is the activation function of all three gates, and produce numbers between “0” and “1” as output. Therefore, a value of 0 (‘off’) will not allow anything to pass through the gate, while a value of 1 (‘on’) will allow everything to pass through the gate. Based on this, the equation of the input gate is given in equation (3), the forget/hidden gate is given in equation (4), and the equation of the exit gate is given in equation (5).

$$i_t = \sigma(X_t V_{xi} + h_{t-1}W_{hi} + C_{t-1}W_{ci} + b_i) \tag{3}$$
$$f_t = \sigma(X_t V_{xf} + h_{t-1}W_{hf} + C_{t-1}W_{cf} + b_f) \tag{4}$$
$$o_t = \sigma(X_t V_{xo} + h_{t-1}W_{ho} + C_{t-1}W_{co} + b_o) \tag{5}$$

V_x, W_h and W_c in the equations are the weight vector associated with X_t, (h_{t-1}) and (C_{t-1}), respectively, while b is the bias weight vector [12]. According to Equation (3), the i_t vector determines how much of the input needs to be updated. The term $X_t V_{xi}$ represents the contribution of the current input to the LSTM cell. On the other hand, the term $h_{t-1}W_{hi}$ represents the contribution of the hidden state to the cell at the previous time. Lastly, the term $C_{t-1}W_{ci}$ represents the contribution of the cell state to the cell at the previous time. The bias term b_i is an additional learnable variable and makes the model more flexible and the sigmoid function σ, is used to determine how much of each entry to update.

As seen in Equation (4), the vector f_t determines how much of the cell state is forgotten. The sigmoid function compresses the output of this gate to a value between 0 and 1, which calculates how much of the cell state will be forgotten. Finally, in Equation (5), o_t determines how much of the cell state is used as output.

In order to calculate the cell state, C_t, the candidate cell state, $\hat{C}_t$ must first be calculated. The candidate cell state is calculated by combining the current input and hidden state vectors. The candidate cell state refers to the potentially updated state of the cell, but not yet filtered by forget and output gates. The candidate cell state and new cell state formula is calculated as follows:

$$\hat{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \tag{7}$$

In Equation 6, $\hat{C}_t$ is the candidate cell state, and this vector is calculated by combining the input vector x_t and the previous hidden state vector h_{t-1}. W_{xc} and W_{hc} in Equation 6 are the weight matrices, while b_c is the bias term. The hyperbolic tangent function tanh limits the candidate cell state and scales the output between -1 and 1.

Once the candidate cell state is calculated, the new cell state can now be calculated. The new cell state is updated by the combination of the previous cell state and the candidate cell state, as given in Equation 7. The f_t in the formula is the output of the forget gate and multiplied by C_{t-1} determines how much of the previous cell state will be forgotten. The i_t is the output of the input gate and is multiplied by $\hat{C}_t$ to determine how much of the candidate cell state will be added to the cell state. The $\odot$ sign in Equation 7 represents element-wise multiplication.

Finally, the hidden state, h_t is calculated. The formula for the hidden state calculated using the output gate and the updated cell state is given in Equation 8. Accordingly, the cell state is passed through the tanh function and subjected to element-wise multiplication with the output gate. Thus, the hidden state is compressed into the (-1, 1) range.

$$h_t = o_t \odot \tanh(C_t) \tag{8}$$

LSTM is well suited for time series forecasting as well as other problems that require temporal memory, as it can learn long-term dependencies without the problem of exploding/vanishing gradients [12]. The key reason for the success of LSTMs in natural language processing lies in their ability to handle long-term dependencies effectively. Sentences in natural language often involve long-term dependencies, where the meaning of a word is strongly related to previous and subsequent words. LSTMs can manage long-term dependencies more effectively through mechanisms such as the forget gate, input gate, and output gate.

3. Results and Discussion

In this study, LSTM, one of the deep learning methods, is used to predict the value of the "Industry that Project Outputs will be used" field. In order to find out which attribute performs better, an Apriori based method was used to train models with different attribute combinations. Model training with only one attribute was performed only for the "Project Name" attribute. The metrics of the model are given in Table 5.

The performance of the model has been evaluated using the metrics Loss, Accuracy, Validation Loss, Validation Accuracy, Test Loss, and Test Accuracy. Accordingly, Accuracy is the ratio of the correctly predicted examples to the total predictions, while Loss measures how erroneous the model's predictions are. Validation Loss indicates the errors made by the model on the validation dataset, whereas Validation Accuracy is the ratio of correctly predicted examples in the validation dataset. Test Loss measures the errors made by the model on the previously unseen test dataset, and Test Accuracy is the ratio of correctly predicted examples in the test dataset [14].

An accuracy of 0.8900 and an error value of 0.4322 were attained, as Table 5 demonstrates. Simply looking at these findings could give the impression that the model operates fairly effectively. However, a notable rise in error and fall in accuracy is seen when looking at the accuracy and error metrics for the Validation and Test sets. This indicates that the model has memorized the data, suggesting overfitting. Indeed, it highlights that the "Project Name" attribute alone has a very weak generalization ability for the model. Therefore, in the next step, other attributes will be used for training alongside the "Project Name" attribute.

Table 5. Metrics of the model trained with the "Project Name" attribute.

Metrics	Values
Loss	0.4322
Accuracy	0.8900
Validation Loss	2.1488
Validation Accuracy	0.2889
Test Loss	2.3862
Test Accuracy	0.2600

Table 6 displays the model's results for each two-element attribute subset that contains "Project Name". As shown in the table, the subset titled "Project Name – Project Summary" obtained the best accuracy and lowest loss rate. Nevertheless, it is noted that in terms of test loss, test accuracy, validation accuracy, and validation accuracy numbers, this combination does not produce the best results. Poor performance on the validation and test set compared to the training set is indicative of overfitting. When these metrics are examined on the test set, it becomes clear that this attribute pair does not effectively satisfy the generalization capability of the model.

Table 6. Metrics of all two-element attribute subsets containing “Project Name”.

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Keywords	0.6935	0.7866	1.8048	0.3630	1.9846	0.3333
Project Name, Scientific and Technological Field of Activity	0.6773	0.8222	1.7980	0.4593	1.7488	0.5200
Project Name, Industry that R&D Studies will be carried out	0.4278	0.8784	2.1168	0.4296	1.9482	0.4600
Project Name, Project Summary	0.4197	0.9007	2.5104	0.3259	2.0752	0.3800
Project Name, Purpose of the Project	0.9899	0.6857	2.2289	0.2963	1.9136	0.2600
Project Name, Outputs of the Project	0.6692	0.7974	2.7115	0.1407	2.3502	0.2733

The "Project Name - Scientific Technological Activity Area" feature combination performed well in both training and test/validation sets, producing more balanced results. With an accuracy value of 0.8222, this combination not only performed quite well, but also produced the highest validation and testing performance among all two-element feature subsets. Therefore, in the next step, when creating three-element subsets, all subsets must contain the pair "Project Name - Scientific Technological Activity Area".

The model metrics for all three-element attribute subsets containing the pair “Project Name – Scientific Technological Activity Area” are given in Table 7. Here, it is observed that all accuracy values in Table 7 yield quite good results compared to the ones in Table 6. However, examining the validation and test values reveals declines similar to those in Table 6.

Table 7. Model metrics for All Three-Element Attribute Subsets Containing the Pair "Project Name – Scientific Technological Activity Area."

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Scientific and Technological Field of Activity, Keywords	0.4130	0.8859	2.2889	0.3407	2.4073	0.3267
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out	0.4828	0.8412	1.7245	0.5259	1.5907	0.5000
Project Name, Scientific and Technological Field of Activity, Summary of the Project	0.5074	0.8395	1.9842	0.3778	2.1492	0.3667

Table 7. Continued

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Scientific and Technological Field of Activity, Purpose of the Project	0.7008	0.8147	2.0090	0.3333	1.8647	0.3933
Project Name, Scientific and Technological Field of Activity, Outputs of the Project	0.6026	0.8420	2.6269	0.2741	2.8459	0.2933

The trio “Project Name, Scientific Technological Activity Area, Keywords” has the lowest error of 0.4130 and the best accuracy of 0.8859 for three-element attribute subsets. Nonetheless, the attribute trio “Project Name, Scientific Technological Activity Area, Industry that R&D Activities are Conducted” yields the best results when it comes to validation and test performance. The power of the model for generalization can be seen more clearly by this trio. Thus, this trio will be employed to generate four-element attribute subsets.

The attribute trio “Project Name, Scientific Technological Activity Area, Industry that R&D Activities are Conducted,” when compared to the pair “Project Name – Scientific Technological Activity Area,” yielded better performance in all metrics except for test accuracy. As the number of attributes increases, the accuracy of the model increases and the error rate decreases. This suggests that the model may be learning more effectively with additional informative attributes, fitting better to the dataset.

The four-element attribute subsets containing the attribute trio “Project Name, Scientific Technological Activity Area, Industry that R&D Activities are Conducted” are provided in Table 8. The highest accuracy value is found in the subset that includes the “Project Purpose” attribute, which surprisingly has the lowest validation accuracy. This suggests that the model may be experiencing an overfitting problem, despite achieving the highest accuracy in this particular subset.

Table 8. Model metrcis for All Four-Element Attribute Subsets Containing the Trio “Project Name – Scientific Technological Activity Area – Industry that R&D Activities are Conducted”.

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Keywords	0.4146	0.8958	1.9712	0.3556	2.1429	0.3667
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Summary of the Project	0.6793	0.8056	2.1528	0.2519	2.1550	0.1800
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Purpose of the Project	0.3823	0.9165	2.5463	0.1852	2.4456	0.2533
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Outputs of the Project	0.6752	0.7891	2.3601	0.2519	2.1120	0.2800

Table 9. Model metrics for All Five-Element Attribute Subsets Containing the Quartet "Project Name – Scientific Technological Activity Area – Industry that R&D Activities are Conducted – Keywords".

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Keywords, Summary of the Project	0.6018	0.8131	1.8527	0.3630	2.0945	0.3600
Project Name, Field of Scientific Technological Activity, Industry that R&D Studies will be carried out, Keywords, Purpose of the Project	0.9220	0.7328	2.3042	0.2963	2.2629	0.2933
Project Name, Field of Scientific and Technological Activity, Industry that R&D Studies will be carried out, Keywords, Outputs of the Project	0.8908	0.7411	2.5708	0.3630	2.4176	0.3533

In general, the performances reflected in Table 8 appear to be significantly lower compared to previous results. The relatively best performance —both in terms of accuracy and validation and test sets— is observed in the subset containing the "Keywords" attribute. Therefore, subsets with five elements will be generated using this attribute set.

In Table 9, model metrics are provided for all five-element attribute subsets containing the quartet "Project Name – Scientific Technological Activity Area – Industry that R&D Activities are Conducted – Keywords". The subset containing the "Project Summary" attribute demonstrates relatively superior performance in terms of accuracy and error values, as well as in test and validation metrics. Therefore, the six-element subsets are constructed based on this quintet subset. The corresponding six-element subsets are also given in Table 10.

Table 10. Model metrics for All Six-Element Attribute Subsets Containing the Quintet "Project Name – Scientific Technological Activity Area – Industry that R&D Activities are Conducted – Keywords – Project Summary".

Combination of Attributes Used	Loss	Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Project Name, Field of Scientific Technological Activity, Industry that R&D Studies will be carried out, Keywords, Summary of the Project, Purpose of the Project	1.0779	0.6311	2.3087	0.1778	1.7953	0.3667
Project Name, Field of Scientific Technological Activity, Industry that R&D Studies will be carried out, Keywords, Summary of the Project, Outputs of the Project	0.1433	0.9760	2.7160	0.2593	2.5852	0.3133

Table 10 shows that, with an accuracy of 0.9760, the subset holding the “Project Outputs” attribute achieves very high model performance. The model is overfitting, nevertheless, as seen by its comparatively poor validation accuracy of 0.2593 and test accuracy of 0.3133.

At last, Table 11 presents the model that was trained utilizing each attribute in the dataset. As seen in Table 11, there is an increase in model accuracy. However, the lack of improvement in the test and validation metrics indicates that the memorization problem encountered in previous models exists here as well.

Table 11. Table contains the metrics of the model trained using all attributes in the dataset.

Metrics	**Values**
Loss	0.3477
Accuracy	0.9330
Validation Loss	2.6620
Validation Accuracy	0.2370
Test Loss	2.9152
Test Accuracy	0.2600

4. Conclusion

In this study, we delved into the critical aspect of feature selection in the context of multi-class text classification. Our study centered around a comprehensive dataset comprising TUBITAK TEYDEB projects, and we sought to unravel the impact of different attribute subsets on model performance. To tackle this challenge, we employed Long Short-Term Memory (LSTM), a deep learning architecture known for its ability to handle sequential data and capture long-term dependencies. Our goal was to classify the projects into nine distinct output industries based on relevant features.

The crux of our approach lay in feature selection. We proposed a novel method grounded in the Apriori algorithm, which systematically pruned the attribute space. By doing so, we aimed to strike a balance between model efficiency and predictive accuracy.

The dataset used in the study contains information about TUBITAK TEYDEB projects. The dataset can be enriched with information obtained from other web sources. This could potentially lead to improved performance. The proposed feature selection method aims to reduce computation costs by selecting new feature sets from the subsets that yields the best performance. However, this does not guarantee finding an optimal feature set. Further research will focus on developing more advanced and effective feature selection methods.

Our findings underscored the pivotal role of feature selection. Among various attribute combinations, the trio comprising “Project Name”, “Scientific Technological Activity Area”, and “Industry that R&D Activities are Conducted” emerged as the most potent. This subset consistently yielded superior model performance across multiple evaluation metrics, including accuracy, loss, validation accuracy, and test accuracy. Our study highlights that judicious feature selection significantly enhances the accuracy and generalization ability of multi-class text classification models. Researchers and practitioners alike should consider these insights when designing robust and efficient systems for real-world applications.

The LSTM model for the multiclass text classification problem is the main topic of this research. There are more sophisticated and recent models, but LSTM is a model that can manage sequential data and long-term dependencies well. It can be utilized in conjunction with these new models in further studies.

Ethical Statement

Our study does not cause any harm to the environment and does not involve the use of animal or human subjects. Therefore, it was not necessary to obtain an Ethics Committee Report.

Conflict of Interest

The authors declare no conflicts of interest.

Authors' Contributions

M.F.E: Investigation, Methodology, Preprocessing, Writing (%60)
T.T.B: Conceptualization, Data collection, Methodology, Formal analysis (%40).
All authors read and approved the final manuscript.

References

[1] Dogra, V., Singh, A., Verma, S., Kavita, Jhanjhi, N. Z., Talib, M. N., Understanding of data preprocessing for dimensionality reduction using feature selection techniques in text classification, in: *Intelligent Computing and Innovation on Data Science: Proceedings of ICTIDS*, Springer, Singapore, pp. 455-464, 2021.

[2] Thirumoorthy, K., Muneeswaran, K., "Feature selection for text classification using machine learning approaches." *National Academy Science Letters*, 45(1), 51-56, 2022.

[3] Amazal, H., Ramdani, M., Kissi, M. (2020). "Towards a feature selection for multi-label text classification in big data." *Proceedings of Smart Applications and Data Analysis: Third International Conference*, Marrakesh, Morocco, June 25–26, 2020, pp. 187-199

[4] Naik, D. A., Mythreyan, S., Seema, S., "Relevance Feature Discovery in Text Mining Using NLP". in: *3rd International Conference for Emerging Technology*, IEEE, pp. 1-6, 2022.

[5] Dowlagar, S., Mamidi, R. "Does a Hybrid Neural Network Based Feature Selection Model Improve Text Classification?", *arXiv preprint arXiv:*2101.09009, 2021. https://doi.org/10.48550/arXiv.2101.09009

[6] Hussain, S. F., Babar, H. Z. U. D., Khalil, A., Jillani, R. M., Hanif, M., Khurshid, K. "A fast non-redundant feature selection technique for text data", *IEEE Access*, 8, 181763-181781, 2020.

[7] Belkarkor, S., Hafidi, I., Nachaoui, M., Feature Selection for Text Classification Using Genetic Algorithm, in: *International Conference of Machine Learning and Computer Science Applications*, Cham: Springer Nature Switzerland, pp. 69-80, 2022.

[8] Zheng, W., "A comparative study of feature selection methods." *International Journal on Natural Language Computing*, 7(5), 01-09, 2018.

[9] Xiaochuan, T., Xiaochuan, T., Yuanshun, D., Yanping, X., "Feature selection based on feature interactions with application to text categorization." *Expert Systems with Applications*, 120, 207-216, 2019. doi: 10.1016/J.ESWA.2018.11.018

[10] TÜBİTAK TEYDEP Proje Değerlendirme ve İzleme Sistemi (2023, Nov. 23). Tamamlanmış Proje Sorgula [Online]. Available: https://eteydeb.tubitak.gov.tr/

[11] re — Regular expression operations. (2024, Jan. 05). Regular expression operations [Online]. Available: https://docs.python.org/3/library/re.html

[12] Van Houdt, G., Mosquera, C., Nápoles, G., “A review on the long short-term memory model”, *Artificial Intelligence Review*, 53(8), 5929–5955, 2020. https://doi.org/10.1007/s10462-020-09838-1

[13] PO, U., Udanor, C. N., Bakpo, F. S., “Deep Learning Algorithms for predicting the geographical locations of Pandemic Disease Patients from Global Positioning System (GPS) Trajectory Datasets”. https://doi.org/10.21203/rs.3.rs-2770308/v1

[14] Rainio, O., Teuho, J., Klén, R. “Evaluation metrics and statistical tests for machine learning”. *Scientific Reports*, 14(1), 6086. 2024.