




# Natural Language Processing in Meaning Representation for Sentiment Analysis in Serbian Language

Marko M. Živanović<sup>1\*</sup>  [0009-0005-9264-3314], Olga Ristić<sup>2</sup>  [0000-0002-1723-0940] and Sandra Milunović Koprivica<sup>2</sup>  [0000-0001-6413-433X]

<sup>1</sup> School of Electrical and Computer Engineering, Academy of Technical and Art Applied Studies, 11000 Belgrade, Serbia

<sup>2</sup> University of Kragujevac/Faculty of Technical Sciences, Čačak, Serbia

\* [markoz@gs.viser.edu.rs](mailto:markoz@gs.viser.edu.rs)

**Abstract:** *This paper explores machine learning algorithms that contribute to meaning representation and context modeling in sentiment analysis. Language preprocessing techniques are described in detail. The study also discusses string distance calculations and the application of Naive Bayes for classification, emphasizing important model metrics such as accuracy. The final section of the paper presents a practical example encompassing the process of data collection, analysis, preprocessing, classification using machine learning algorithms, and model evaluation. Testing demonstrated the system's ability to classify sentiments in Serbian Language.*

**Keywords:** *Naive Bayes; Model Metrics; Meaning; Context; Sentiments*

## 1. INTRODUCTION

In recent decades, the world has witnessed an explosion of textual data from various platforms, ranging from social media to blogs and news articles. This influx of data has necessitated advanced methods for processing and analyzing it, driving the development of Natural Language Processing (NLP). NLP has become crucial for understanding and exploiting textual data. Initially, NLP deals with text preprocessing, which includes processes such as tokenization, normalization, and lemmatization, essential for preparing the text for further analysis. These processes serve as the foundation for applying various machine-learning algorithms that enable the extraction of meaning from text. Besides preprocessing, NLP encompasses the development of models capable of understanding and generating text. In this context, machine learning algorithms are of paramount importance. From Naive Bayes classifiers to advanced techniques, these models facilitate sentiment analysis, emotion recognition, and context modeling. This paper explores a dataset collected from confession forums, aiming to apply machine learning algorithms for sentiment classification in confessions. The first step in the research is to analyze the distribution of sentiments in the data using a classification based on the number of approvals and disapprovals. Following this, the WordCloud tool is used to examine the most prevalent words in the confessions. The

accuracy of the classifiers in different languages is also analyzed, and the challenges of processing text in multiple languages are considered. Through various metrics and analyzes, this paper provides deeper insights into the sentiments and emotional states expressed in the confessions. Finally, the most successful classifiers for sentiment analysis in different languages are identified, and recommendations for future research in the field of NLP based on these confessions are provided.

## 2. RELATED WORK

In recent years, the development and application of NLP techniques in sentiment analysis have gained significant attention, particularly in less-resourced languages such as Serbian. Sentiment analysis involves the extraction and analysis of subjective information from textual data, which is crucial for various applications.

The task of sentiment analysis in the Serbian language presents unique challenges due to its linguistic characteristics, including the complex morphology which can significantly influence the overall sentiment value of a text. Jahić and Vičić [14] explored these challenges in the context of the Bosnian language, a closely related South Slavic language, highlighting the impact of negation on sentiment classification. Their findings underscore the importance of addressing these linguistic features in sentiment analysis models to enhance their accuracy and reliability.

Building on the foundation of multilingual sentiment analysis, Draskovic et al. [15] developed a multilingual model for machine sentiment analysis specifically tailored to the Serbian language. Their research demonstrated the potential for cross-linguistic approaches in enhancing sentiment analysis capabilities for Serbian, emphasizing the need for models that can effectively capture the nuances of the language across different domains.

Further advancements in NLP for the Serbian language are illustrated by Laković et al. [16], who conducted an exploratory analysis of text using available NLP technologies. Their study provided insights into the current state of NLP tools for Serbian, highlighting both the opportunities and limitations in applying existing technologies to sentiment analysis tasks. The results of their work suggest a need for continued refinement and adaptation of NLP tools to better serve the specific requirements of Serbian language processing.

One of the significant contributions to Serbian NLP is the development of SRBerta, a Transformer-based language model specifically designed for Serbian Cyrillic legal texts by Bogdanović et al. [17]. While their work primarily focused on legal documents, the underlying technology of SRBerta holds promise for broader applications, including sentiment analysis. The use of such advanced models represents a critical step forward in the creation of robust, domain-specific NLP tools for Serbian, capable of handling the intricacies of the language in various contexts.

Sentiment analysis for the Serbian language illustrates a growing interest and progress in the field, driven by the development of both general and domain-specific models. These efforts reflect the ongoing challenges and opportunities in adapting NLP techniques to effectively process and analyze text in the Serbian language.

### 3. MEANING REPRESENTATION IN THE FIELD OF NLP

Meaning representation in NLP is the process of representing words, sentences, or text in a way that allows computers to understand and implement systems. It involves capturing information or concepts as they are represented in the human brain or in a computational system. Meaning representation is crucial in various NLP applications, including machine translation, sentiment analysis, text classification, text generation, automatic detection of inappropriate content, detection of emotional tone in text, and automatic identification of text authorship.

A collection of text or speech that a computer can process and read is called a corpus. Sentences contain words and punctuation marks. Punctuation determines the boundaries of elements and uses marks like question marks and quotation marks to

identify context. Punctuation is highly significant in sentiment analysis. Sometimes sentences contain words that are not important for a particular system, such as interjections, which can be excluded. The distinction between capitalized and lowercase words depends on the task; for tasks like speech representation and recognition, it is not very important. According to "Serbian Dictionary" by Vuk Stefanović Karadžić [1], there are 26,270 words in the Serbian literary language. The dictionary of the Serbian Academy of Sciences and Arts predicts it will have over 35 books with about 500,000 entries [2]. However, the exact number of words is hard to determine due to loanwords, archaisms, etc.

The larger the corpus analyzed, the more word types we find. The number of word types is denoted as  $|V|$ , and the number of occurrences  $N$  is defined by Herdan's law with the formula (1).

$$|V| = kN^\beta \quad (1)$$

Where  $k$  and  $\beta$  are positive constants within  $0 < \beta < 1$ . The value of  $\beta$  depends on the corpus and genre, and the vocabulary size significantly increases, more than the square root [3].

#### 2.1. Lemmatization

*Lemmatization* is the task of determining whether two words have the same root despite all differences. For example, the holiday „Spasovdan“ has the root word „spas“ (salvation) in its name. Lemmatization involves the complete morphological parsing of words. Words consist of smaller units called morphemes, which carry meaning. For instance, the word „knjiga“ (book) or „knjige“ (books) has the lemma „knjiga“, which is the base form. Significantly, words are morphologically different because they have singular and plural forms. For example, in the name of the town „Bratunac“, lemmatization would yield the words „brat“ (brother) and „unac“.

#### 2.2. Stemming

Trimming the endings of words is called stemming. Porter introduced a precursor to stemming in his work from 1980 [4]. An example of stemming according to Porter would be as in the text:

*„Постоји још неколико теорија о настанку имена Братунац. Наиме, према причи старијих становника општине Братунац, име Братунац је настао тако што је неки бег, који је био најзначајнији у то вријеме на том простору звао свог сестрића: „Ходи амо, дајци мој братунац...“, (син од сестре-братанац-братунац).“[5]*

This text would look like this:

*“Постоји још неколико теорија о настанку име Братунац. Наиме, према причи стариј становник општин Братунац, им Братунац је наста так што је неки бег, кој је био најзначајни у то вријем на том простор зва свог сестрић: „Ход амо, дај мој*

братунац...", (син од сестре-братанац-братунац)."

### 2.3. Minimal distance changes in strings

The distance in changes provides a way to express sentences at different distances. The minimal distance between two arrays gives the minimum number of change operations (insertion, deletion, substitution) required to convert one array into another.

Levenshtein distance or edit distance is a measure used to calculate the similarity between two character arrays, also known as cost. It is defined as the minimum number of simple operations required to transform one character into another. The operations are inserting a new character, deleting existing characters, or replacing one character with another [6].

Levenshtein distance between the words „Сребро“ and „Сребреница“ is 5 because it requires 5 insertions of 'еница' after the word сребро.

- Сребро -> Сребре – Transition o to e
- Сребре -> Сребрен – Adding the letter н after e
- Сребрен -> Сребрени – Adding и after н
- Сребрени -> Сребрениц – Adding ц after и
- Сребрениц -> Сребреница – Adding а at the end

The distance between the words „ПОДРИЊЕ“ and „ПОДРИЊСКИ“ is 1.

- ПОДРИЊЕ -> ПОДРИЊС – Transition e to c
- ПОДРИЊС -> ПОДРИЊСК – Adding the letter к after c
- ПОДРИЊСК -> ПОДРИЊСКИ – Adding the letter к after и

The distance between the words Братанац and Братунац is 1.

БРАТУНАЦ -> БРАТАНАЦ"

### 2.4. Naive Bayes

The Naive Bayes classifier is a probabilistic classification model that applies Bayes' rule. The basic idea is represented by formula (2), where  $d$  from the class set  $C$  is assigned to the class with the highest posterior probability [7].

$$\hat{C} = \arg \max_{c \in C} P(c|d) \quad (2)$$

By applying Bayes' rule, we arrive at formula (3), which maps the posterior probability based on the model and prior probability. Of course, these formulas make assumptions about the independence of word identity, i.e., that the position of words is not important, which is a characteristic of the naive Bayes approach [8].

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \quad (3)$$

The training process of the naive Bayes classifier is crucial for probability modeling. Formula (4) represents the prior probability of class  $c$ , where

the assumption of maximum probability is applied [9].

$$P(c) = \frac{N_c}{N_{doc}} \quad (4)$$

Training involves calculating probabilities based on the training dataset, but Laplace's formula (5) is used to avoid problems with zero probabilities when data is sparse [10].

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i,c)+1}{(\sum_{w \in V} \text{count}(w,c))+|V|} \quad (5)$$

By incorporating word positions, formula (6) further modifies the classification process.

$$\hat{C}_{NB} = \arg \max_{c \in C} P(c) \prod_{i \in \text{positions}} P(w_i|c) \quad (6)$$

Finally, to represent documents as a set of features, formula (7) is used.

$$\hat{c} = \arg \max_{c \in C} P(f_1, f_2, \dots, f_n |c) P(c) \quad (7)$$

### 2.5. Application of Naive Bayes in Sentiment Analysis

In the example in Table 1, different sentences are given for classification using Naive Bayes.

**Table 1.** The example values are given in column X, and the classes are given in column Y.

X	Y
Братунац бисер Републике Српске.	1
Сребреница град са богатом културном баштином.	1
Подриње, представља идеално место за релаксацију и одмор.	1
Братунац, место са трагичном претходном историјом.	0
Сребреница, име које је постало симбол масовног страдања.	0
Подриње напомиње на последице ратних сукоба.	0
Подриње је познато по рату.	0
<b>Сребреница бисер источне Српске.</b>	<b>?</b>

Based on Table 1, we conclude that the ratio of probabilities of positive elements is given by formula (8), and the ratio of probabilities of negative elements is given by formula (9).

$$P(\text{positive}) = \frac{3}{7} \quad (8)$$

$$P(\text{negative}) = \frac{4}{7} \quad (9)$$

According to the formula, the total number of words in the corpus is determined as:

$$|V| = 43 - 7 \quad (10)$$

Free examples of sentiment analysis on forums are given in Table 2.

**Table 2.** Statistical values of words in the corpus

Query	Value (corpus in %)
подриње	3 / 7
братунац	2 / 4.7
сребреница	2 / 4.7
са	2 / 4.7
место	2 / 4.7
је	2 / 4.7
<b>Positive words</b>	<b>18</b>
<b>Negative words</b>	<b>25</b>
<b>Sum of all words</b>	<b>43</b>

In the sentence  $S = \text{"Сребреница бисер источне Српске"}$ , the word "источна" is not in the training corpus and that word is removed.

The procedure for determining the probability of whether a sentiment is positive or negative is given by formulas: (11 - 18).

$$P(\text{"Сребреница"}|pos) = \frac{1+1}{18+36} \quad (11)$$

$$P(\text{"Сребреница"}|neg) = \frac{1+1}{25+36} \quad (12)$$

$$P(\text{"бисер"}|pos) = \frac{1+1}{18+36} \quad (13)$$

$$P(\text{"бисер"}|neg) = \frac{0+1}{25+36} \quad (14)$$

$$P(\text{"Српске"}|pos) = \frac{1+1}{18+36} \quad (15)$$

$$P(\text{"Српске"}|neg) = \frac{0+1}{25+36} \quad (16)$$

$$P(neg)P(S|neg) = \frac{4}{7} \times \frac{2 \times 1 \times 1}{61 \times 61 \times 61} \approx 4.707 \times 10^{-8} \quad (17)$$

$$P(pos)P(S|pos) = \frac{3}{7} \times \frac{2 \times 2 \times 2}{54 \times 54 \times 54} \approx 7.742 \times 10^{-7} \quad (18)$$

For sentiment analysis, the standard Naive Bayes text classification can be adapted to improve performance. The key modification is to focus on the presence of a word in the document rather than its frequency. **Limiting the number of occurrences of a word to 1 in each document can significantly improve results. This preprocessing is known as binary multinomial Naive Bayes**, which is based on the same algorithm as the regular Naive Bayes. During training, duplicate words are removed before merging into a single document, both in the training and test documents.

Negation plays a significant role in sentiment analysis, for example, "Дринска регата ми се свидела" (I liked the Drina Regatta) versus "Дринска регата ми се није свидела" (I did not like the Drina Regatta). **Negation completely changes the conclusion and gives a different output.** In some algorithms where negation appears, such as "није" (is not), "не" (no), "никако" (by no means), "никакво" (no kind of), and similar words, the particle НЕ (NO) is usually added in place of these words!

## 2.7. Model Metrics in Machine Learning

The basic metrics for evaluating machine learning models are: Accuracy, Precision, Recall, and F1-Score.

**Accuracy** - This is a general measure of model performance, defined as the ratio of correctly classified instances (TP and TN) to the total number of instances (19). Accuracy is useful when the classes in the dataset are relatively balanced but can be misleading otherwise [11].

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (19)$$

**Precision** - This metric measures the accuracy of the model's positive predictions, i.e., the ratio of correctly predicted positive instances to all instances that the model labeled as positive (TP / (TP + FP)) (20). Precision is important in scenarios

where minimizing the number of false positive predictions is crucial, such as in medical diagnostics [11].

$$Precision = \frac{TP}{TP+FP} \quad (20)$$

**Recall:** Also known as sensitivity or the true positive rate (TPR), recall measures the model's ability to identify all actual positive instances in the dataset (TP / (TP + FN)) (21). High recall indicates that the model rarely misses positive instances, which is important in tasks where it is critical to find all positive examples [11].

$$Recall = \frac{TP}{TP+FN} \quad (21)$$

**F1 Score:** This measure represents the harmonic mean between precision and recall, and is useful when a balance between these two metrics is needed. The formula for the F1 score is  $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$  (22). A high F1 score indicates a good balance between precision and recall [11].

$$F = \frac{1}{\left(\alpha \frac{1}{Precision} + (1-\alpha) \frac{1}{Recall}\right)}, \alpha \in [0,1] \quad (22)$$

When the parameter  $\alpha < 1/2$ , the formula emphasizes recall; when  $\alpha = 0$ , the F-measure is equal to recall, specifically  $F = Recall$ . When  $\alpha > 1/2$ , the formula highlights precision; for  $\alpha = 1$ ,  $F = Precision$ . In the case where  $\alpha = 1/2$ , the formula represents the harmonic mean of precision and recall and is referred to as the balanced measure.

**This measure is also referred to as the Macro F1 measure.** (23)

$$F = \frac{2 * Precision * Recall}{Precision + Recall} \quad (23)$$

## 2.8. ROC/AUC curve

The ROC (**Receiver Operating Characteristic**) **curve** is a graphical representation of classifier performance, illustrating the relationship between **True Positive Rate (TPR)** and **False Positive Rate (FPR)** for different decision thresholds. It is based on binary classification problems. TPR, also known as sensitivity, represents the ratio of true positives to the total number of positive instances in the dataset. It is given by the expression (24).

$$TPR = \frac{TP}{TP+FN} \quad (24)$$

FPR represents the ratio of false positives to the total number of negative instances in the dataset. The ROC curve is a graph where the x-axis represents FPR and the y-axis represents TPR [12]. The closer the ROC curve is to the upper-left corner of the graph, i.e., closer to the ideal case (TPR=1, FPR=0), the better the classifier. It is given by the expression (25).

$$FPR = \frac{FP}{TN+FP} \quad (25)$$

AUC (Area Under the Curve) represents the area under the ROC curve and is used as a measure of classifier performance. The AUC value ranges between 0 and 1, where a value of 1 represents the ideal situation (perfect classifier), while a value of

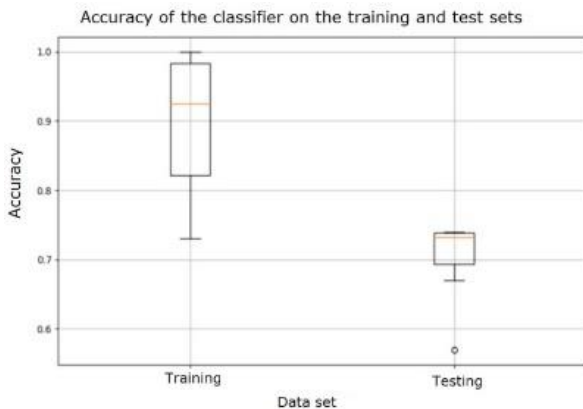
0.5 indicates random guessing. The closer the AUC is to 1, the better the classifier performs. AUC can also be used to compare multiple models [12].

**2.9. Boxplot**

Boxplot is a graphical representation of data distribution that enables visual analysis of the statistical characteristics of a dataset. This graph is used to display the distribution of numerical data through five basic statistical parameters: minimum, first quartile (25th percentile), median (50th percentile), third quartile (75th percentile), and maximum [13].

The main purpose of a boxplot is to provide insight into the central tendency, dispersion, and presence of any extreme values in the dataset. The central line in the boxplot represents the median (the value that divides the dataset into two halves of 50% each), while the lower and upper edges of the rectangle (the "box") represent the first and third quartiles, respectively. Vertical lines extending from the boxes indicate the range of data values outside the interquartile range, and any points beyond these lines represent extreme values or outliers.

Boxplots are useful for comparing distributions of different datasets or for analyzing changes in data distribution over time or between different groups. Additionally, they assist in identifying any outliers and provide insight into the symmetry or asymmetry of the data distribution. Boxplot visualization on the training and testing datasets for sentiment analysis in the Serbian language (Fig 1).



**Figure 1.** Boxplot visualization on the training and testing data sets

**2.10. Matrix confusion**

Evaluation of machine learning models provides a detailed overview of the performance and quality of machine learning algorithms. To evaluate a machine learning model, an appropriate model metric must be chosen. In machine learning classification for spam detection, accuracy is often chosen as the metric; however, in some situations, this metric may not be the most suitable.

The evaluation of a machine learning model is conducted exclusively on the test dataset, never on the training dataset. For example, in sentiment detection on social networks, a binary classifier is used to classify as spam or not spam.

The first step in model evaluation is the confusion matrix. The confusion matrix is mainly used in classification tasks (Table 3). The confusion matrix is a square matrix used for model evaluation and has the following dimensions as in equation (26):

$$\text{Dim}_{\text{confusion\_matrix}} = N \times N \quad (26)$$

The confusion matrix provides a detailed insight into how the model classifies data and how well it performs compared to the actual values. The matrix is used in binary classification but can also be used in multiclass classification. The matrix compares the actual target values with the values predicted by the machine learning model.

**Table 3.** Confusion matrix 2 x 2

		Actual values	
		positive	negative
Predicted values	positive	<b>TP</b>	<b>FP</b>
	negative	<b>FN</b>	<b>TN</b>

The confusion matrix is commonly used in sentiment detection with two possible outcomes: whether the sentiment is positive or negative. It is defined as a 2x2 square matrix with the following possible outcomes:

**True Positives (TP):** The number of sentiments correctly classified as positive (predicted value is the same as the actual value).

**False Positives (FP):** The number of instances incorrectly classified as positive (sentiments that are negative but classified as positive). The predicted value is incorrectly predicted.

**True Negatives (TN):** The number of instances correctly classified as negative (predicted negative value is the same as the actual negative value).

**False Negatives (FN):** The number of instances incorrectly classified as negative (positive sentiments incorrectly marked as negative). The predicted value is incorrectly predicted.

Based on these four values, various performance evaluation metrics for the model can be calculated, such as Accuracy, Precision, Recall, and F1-Score.

For example, the dataset contains 5000 instances to be classified, primarily confessions from the website. The dataset is shown in Table 4. When the model is defined and trained, the confusion matrix obtained from the example is significantly imbalanced. There is a much larger number of correctly classified positive sentiments compared to

correctly classified negative sentiments. The minority class is negative, and the majority class is positive. When the data is balanced by classes, we obtain accuracy as the model metric.

**Table 4.** Dataset with the distribution of values

<b>3851</b>	<b>152</b>
<b>54</b>	<b>943</b>

Accuracy is calculated using formula (27)

$$Accuracy = \frac{3851+943}{3851+152+54+943} \approx 0,96\% \quad (27)$$

When, as in the given example, there are many more data points of one class than the other, accuracy ceases to be a useful metric. When working with imbalanced data, the most commonly used measure is the **F-measure**. This measure is defined as the harmonic mean of precision and recall.

From this model, it can be seen that the accuracy is approximately 96%, which indicates that the accuracy is 96% regardless of whether the sentiment is positive or negative. Based on accuracy, there is no information on how many of the total number of actual positive sentiments are correctly predicted to be positive.

Given the seriousness of the problem, accuracy is not a good metric in this case, and another metric needs to be used. The calculated accuracy from the table is given by formulas (28) and (29) respectively:

$$Precision = \frac{3851}{3851+152} = 0,962 \approx 0,96 \quad (28)$$

$$Recall = \frac{3851}{3851+54} = 0,986 \approx 0,99 \quad (29)$$

The Macro F1 score is the harmonic mean of the precision and recall, calculated separately for each class and then averaged:

$$F1 = 2 \times \frac{0,96 \times 0,99}{0,96 + 0,99} \approx 0,974 \quad (30)$$

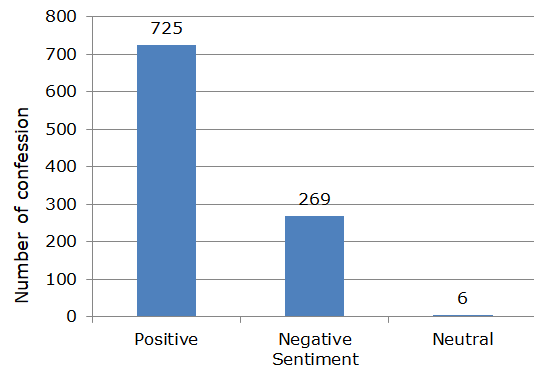
The Micro F1 score aggregates the contributions of all classes to compute the average metric. For binary classification, the Micro F1 score is the same as the accuracy, precision, and recall when they are calculated across all instances:

$$Micro\ F1 = 2 \times \frac{0,962 \times 0,986}{0,962 + 0,986} \approx 0,975 \quad (31)$$

#### 4. DATA COLLECTION AND IMPLEMENTATION OF PRACTICAL EXAMPLE

The popular website "Confessions" <https://ispovesti.com/> where users can anonymously share their personal stories, secrets, confessions, and reflections. The most significant features of the website include anonymity, confessions, voting, comments, categories, search, popularity, and the latest news. The dataset consists of 1000 confessions downloaded from the Confessions website, and the main characteristic is that the dataset itself comprises various emotional

states, punctuation marks, emoticons, and the like. The downloaded dataset has sentiment distribution and a target column determined based on the "approve" and "condemn" fields. The distribution of sentiments is shown in Fig 2.



**Figure 2.** Distribution of sentiments

**WordCloud** is a visualization technique that displays words from a text dataset, with the size of each word indicating its frequency or importance within the dataset. The word cloud visualization for positive sentiments is shown in Fig 3 and in Fig 4 is the word cloud for negative sentiments.



**Figure 3.** The Word Cloud in Positive Classes



**Figure 4.** The Word Cloud in Negative Classes

In Table 5 various results for the model metric are presented. This may indicate overfitting of the model to the training set, which is a common issue when working with machine learning. The best models, such as Random Forest and Decision Tree, should be further investigated to determine



whether their performance on the test set is truly based on their general capabilities or if the results are due to overfitting.

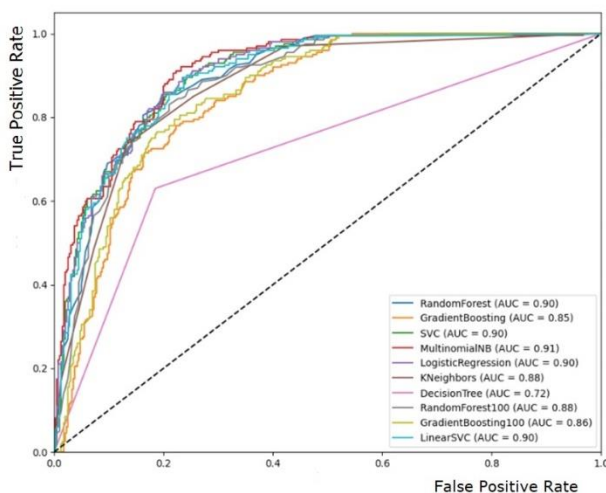
**Table 5.** Training and test set values

Model metric	Training	Test
Random Forest	1.0	0.73
Gradient Boosting	0.92125	0.675
SVC	0.9275	0.74
Multinomial NB	0.73	0.74
Logistic Regression	0.82125	0.72
K-Neighbors	0.765	0.735
Decision Tree	1.0	0.56
Random Forest 100	1.0	0.73
Gradient Boosting 100	0.9225	0.685
Linear SVC	0.82375	0.74

ROC/AUC curves and solutions obtained for the Serbian language. For each language group, the best model that provides the highest AUC can be identified, while the **Decision Tree** classifier is less effective in all analyzed cases. This information can be useful in selecting models for specific language tasks in the future. **Decision Tree** has the weakest performance. For the Serbian language, the AUC is 0.72. **SVM** and **Logistic Regression** classifiers show good results for the Serbian language, while **Decision Tree** shows significantly worse results compared to other classifiers.

The **poor decision** to use the Decision Tree model can be attributed to its tendency for **overfitting, instability, and limited** ability to model complex patterns in the data.

Multinomial NB achieves the best result for Serbian language of 0.91 AUC, as shown in Fig 5.



**Figure 5.** ROC curve for different classifiers

The overview of values provided is located in the lines below:

- **RandomForest: AUC = 0.90**
- **GradientBoosting: AUC = 0.85**
- **SVC: AUC = 0.90**
- **MultinomialNB: AUC = 0.91**
- **LogisticRegression: AUC = 0.90**

- **KNeighbors: AUC = 0.88**
- **DecisionTree: AUC = 0.72**
- **RandomForest100: AUC = 0.88**
- **GradientBoosting100: AUC = 0.86**
- **LinearSVC: AUC = 0.90**

**5. CONCLUSION**

Based on a detailed and systematic approach to processing and analyzing textual data, this paper provides significant insights into the field of NLP and its potential for understanding, interpreting, and exploiting textual information. By applying a wide range of text preprocessing techniques, including tokenization, normalization, and lemmatization, the paper demonstrates crucial methodological aspects necessary for effective textual data analysis. Notably, the use of various metrics to evaluate the performance of machine learning models is emphasized. Precision, recall, F-measure, ROC/AUC curve, and other metrics enable a comprehensive analysis and assessment of different models' performances, which is essential for selecting the best model for a specific application.

Additionally, by exploring machine learning algorithms for sentiment classification in confessions from the "Confessions" website, the paper provides insights into various aspects of sentiment and emotional state modeling in text. Through the application of different metrics and analytical methods, the paper achieves a deeper understanding of sentiment in confessions and identifies the most effective classifiers for sentiment analysis in the Serbian language.

The paper not only contributes to the advancement of methods and techniques in NLP but also paves the way for future research in this field, offering valuable recommendations for further investigation and development in NLP. We conclude that in order to achieve better performance we have to proceed with enlarging dataset for model training.

One of the most significant constraints in this research is the linguistic complexity of the Serbian language. The availability of high-quality, large-scale datasets in the Serbian language is limited. Most existing datasets are either too small for robust machine learning model training or not specifically designed for sentiment analysis tasks. This limitation affects the generalizability and performance of the models trained on such data. The application of Naive Bayes algorithm can lead to lower accuracy and precision in classification tasks, particularly when dealing with complex sentence structures or context-dependent sentiments. This constraint limits the ability to experiment with and deploy more sophisticated models that could potentially offer better performance.

## ACKNOWLEDGEMENTS

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of Grant No. 451-03-66 / 2024-03 / 200132 with the University of Kragujevac - Faculty of Technical Sciences Čačak.

## REFERENCES

- [1] Караџић Стефановић, В. (1818). *Српски Рјечник*. Штампарија Јерменског манастира.
- [2] Институт за српски језик САНУ. (n.d.). Retrieved Februar 23, 2024, from <https://web.archive.org/web/20190323133841/http://www.isj.sanu.ac.rs/projekti/rsanu/>
- [3] Herdan, G. (1960). *Type-token mathematics*. Mouton.
- [4] Porter, M. F. (1980). *An algorithm for suffix stripping*. *Program*, 14(3), 130-137.
- [5] Западни Срби. (n.d.). *Сарајевска регија и Романија*. Retrieved from March 03, 2024, from <https://www.zapadnisrbi.com/zapadni-srbi/republika-srpska3/sarajevsko-romanijska-regija/20-sarajevska-regija-i-romanija?showall=1>
- [6] Levenshtein, V. I. (1966). *Binary codes capable of correcting deletions, insertions, and reversals*. *Soviet Physics Doklady*, 10(8), 707-710.
- [7] Mosteller, F., & Wallace, D. L. (2012). *Applied Bayesian and classical inference: The case of the Federalist papers*. Springer Science & Business Media.
- [8] Mosteller, F., & Wallace, D. L. (2012). *Applied Bayesian and classical inference: The case of the Federalist papers*. Springer Science & Business Media.
- [9] Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing*. <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf> Last access to the site: 30<sup>th</sup> of May 2024.
- [10] Peng, F., Schuurmans, D., & Wang, S. (2004). Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, 7, 317-345.
- [11] Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. *IEEE Access*, 10, 19083-19095.
- [12] Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3), 299-310.
- [13] Dawson, R. (2011). How significant is a boxplot outlier?. *Journal of Statistics Education*, 19(2).
- [14] Jahić, S. & Vičić, J. (2023). Impact of Negation and AnA-Words on Overall Sentiment Value of the Text Written in the Bosnian Language. *Applied Sciences*, 13, 7760. <https://doi.org/10.3390/app13137760>
- [15] Draskovic, D., Zecevic, D. & Nikolic, B. (2022). Development of a Multilingual Model for Machine Sentiment Analysis in the Serbian Language. *Mathematics*, 10, 3236. <https://doi.org/10.3390/math10183236>
- [16] Laković, L., Čakić, S., Jovović, I. & Babić, D. (2023). Exploratory analysis of text using available NLP technologies for Serbian language, *22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, pp. 1-4, <https://doi.org/10.3390/10.1109/INFOTEH57020.2023.10094202>
- [17] Bogdanović, M., Kocić, J. & Stoimenov, L. (2024). *SRBerta—A Transformer Language Model for Serbian Cyrillic Legal Texts*. *Information*, 15, 74. <https://doi.org/10.3390/info15020074>