Language Detection using Natural Language Processing

Dr. A V Sriharsha[1*], Muthyala Reddy Jahnavi[2], Desai Sakethram Kousik[3], Vukyam Hemanth[4],Matchandrappa Gari Hari[5,] Penchala Praveen Vasili[6]

[1] Professor, Department of CSE(DS), Mohan Babu University
(Erstwhile Sree Vidyanikethan Engineering College), India
[2, 3, 4, 5]UG Scholar, Department of Computer Science and Systems Engineering,
Sree Vidyankethan Engineering College, Tirupati, India.
[6]Product Manager, Wellsfargo Inc. Charlotte, USA

[1*] avsreeharsha@gmail.com, [2] reddyjahnavi3013@gmail.com
[3] sakethdesai220503@gmail.com, [4] vukyamhemanth12@gmail.com,
[5]hariediga1050@gmail.com, [6]penchalapraveen@gmail.com

Abstract. Natural Language Processing (NLP) is a rapidly advancing field of artificial intelligence that actsas a bridge between human language and machines. Its uses vary from language translation andsentiment analysis to virtual assistants, impacting a wide range of industries. Language detection is a crucial sub-task of NLP that automatically recognizes the language in a given text. The Mul- tinomial Naive Bayes classifier's effectiveness and performance in text classification, along with NLP feature engineering, make it a suitable option for language detection tasks, even when work-ing with multilingual datasets. By integrating NLP techniques and the Multinomial Naive Bayes classifier, the proposed method offers a strong and precise language detection approach. Exper-iments conducted on diverse textual data show promising outcomes, even when dealing with noise and incomplete information. Accurate language identification improves the usability and efficiency of various NLP applications, promoting cross-cultural communication and contrib- uting to a more inclusive and interconnected digital environment.

Keywords: Natural Language Processing (NLP), Multinomial Naive Bayes classifier, Artificial intelligence, Language translation, Performance evaluation, Text classification.

1 Introduction

Language detection is a crucial task in the field of natural language processing (NLP) and holds significant importance in a wide range of applications that deal with textual data. The main objective of language detection is to automatically determine the language of a given file or document, enabling more precise and contextually aware processing. Accurate language detection is essential in a multilingual world where content filtering, user localization, and information retrieval are increasingly important. The core principles of language detection involve the use of statistical and machine learning methods to distinguish one language from another based on textual features. The capability of deep learning to automatically learn and extract significant patterns from text data has led to increasing popularity as a subset of machine learning. Multinomial Naive Bayes, a probabilistic algorithm, is well-suited for language detection tasks by modeling the probability distribution of words in different languages. Combining deep learning and Multinomial Naïve Bayes offers a robust framework for accurate language identification. Language detection has a wide range of practical applications. Content filtering helps identify and categorize content according to lan- guage preferences, ensuring that users receive content in their preferred languages. User locali zation

K. R. Madhavi et al. (eds.), *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024)*, Advances in Computer Science Research 112,
https://doi.org/10.2991/978-94-6463-471-6_49

search results and targeted advertising. Information retrieval systems use language detection to retrieve documents in the user's preferred language, improving search relevance. Despite its significance, language detection presents several challenges. Code-switching, where multiple languages are mixed within a single text, poses a substantial challenge for accurate identification. Dialect variations within a language further complicate the task. Moreover, the increasing prevalence of multilingual content on the internet requires robust language detection models that can handle diverse linguistic contexts. Addressing these challenges is crucial for advancing the field of language detection and enabling more precise and adaptable language identification systems..

2 Proposed Work

MNB is chosen for language detection because it operates within a probabilistic framework. It models the probability distribution of words or character sequences in different languages. This is crucial because language identification involves making probabilistic assessments based on linguistic patterns. Different languages have unique linguistic characteristics and patterns. MNB is effective in capturing these language-specific patterns by analyzing the frequency of words or characters in a text. Probability calculations are employed to ascertain the probability of a particular language being the origin of a given text. MNB is adaptable to different languages and text types. It can handle a wide range of linguistic features, making it suitable for identifying languages in diverse contexts, from formal documents to user-generated content like social media posts and micro-blogs. MNB is a type of statistical classifier, Multinomial Event Model, and Bayesian Inference. MNB is a statistical classifier, specifically designed for text classification tasks. It is used to classify documents or text snippets into predefined categories or classes, with language identification being a prominent example. MNB is tailored for discrete data, such as word frequencies or term occurrences in text. It employs the multinomial event model, which is well-suited for text classification. In the context of language detection, it categorizes text-based language on the occurrence and frequency of words or characters. MNB's "Naive Bayes" component refers to its reliance on Bayes' theorem for probability calculations. The observed word frequencies are used to calculate the posterior probability of a text belonging to a specific language class. The "naive" assumption in Naive Bayes is that features (words or characters) are conditionally independent, simplifying the probability calculations.

CREMA-D. These datasets were chosen due to their diverse emotional content and the availability of labelled emotional categories. TESS comprises recordings of North American English speakers portraying seven emotional states, while RAVDESS contains speech samples from actors representing eight emotional categories. SAVEE consists of British English speakers expressing seven emotions, and CREMA-D features audio clips from actors portraying various emotions in a controlled environment.

.

3 Algorithms

A. MULTINOMIAL NAIVE BAYES:

MNB, a widely employed algorithm in Natural Language Processing (NLP), is renowned for its simplicity and efficacy in text classification assignments. It operates under a probabilistic framework, making it suitable for language detection. In language identification, MNB calculates the likelihood of a given text belonging to each possible language category and selects the language with the highestprobability. It is often trained on a dataset containing text samples from various languages, al- lowing it to learn language-specific patterns

B. Support Vector Machines (SVM):

SVMs are known for their ability to find a hyperplane that maximizes the separation between different language categories. They can be effective for language detection when coupled with appropriate feature representations. MNB vs. SVM: MNB is computationally more efficient and simpler to implement, which can be advantageous in real-time language detection scenarios. SVMs may require more computational resources and careful parameter tuning.

C. Decision Trees:

Decision Trees create a tree-like structure to classify text based on features. They are interpretable and can handle language detection tasks but may struggle with complex language-specific patterns. MNB vs. Decision Trees: MNB typically excels in capturing subtle language-specific patterns through probabilistic modeling, making it more robust for language detection.

D. K-Nearest Neighbors (KNN):

KNN relies on similarity measures to assign a language to text. It considers the majority class among its nearest neighbors. MNB vs. KNN: MNB probabilistic approach often outperforms KNN, especially when dealing with diverse linguistic contexts, as it captures more nuanced language patterns

E. Ensemble Methods:

Ensemble methods combine multiple classifiers, such as MNB, SVM, or Decision Trees, to improve overall accuracy through techniques like majority voting. MNB vs Ensemble Methods: MNB can be a vital component of ensemble methods, contributing its probabilistic strength to enhance accuracy. However, ensemble methods may introduce computational overhead

4 PROPOSED METHODOLOGY

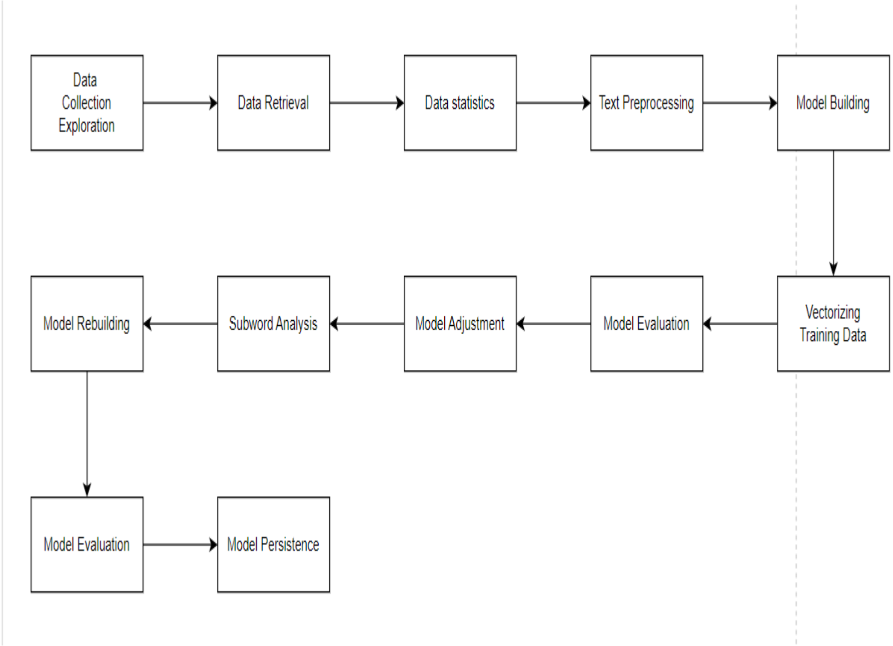

Fig 1: Framework of Text Evaluation

The framework presented outlines a comprehensive process for developing and refining a Multinomial Naive Bayes model for language classification. It begins with data collection and exploration, where training and validation data for Slovak, Czech, and English languages are gathered and statistically analyzed. Subsequent steps involve data cleaning and preprocessing to ensure consistency, followed by the construction of the initial model using the Count Vectorizer for numerical conversion of text data. Model evaluation is conducted using standard metrics like confusion matrix and F1 score. Fine-tuning techniques are applied to enhance model accuracy, particularly for less common languages. The framework then introduces sub-word analysis to further improve performance by identifying common character sequences within words. The model is rebuilt and evaluated using sub-word-based data, and the final trained model along with the Count Vectorizer is saved for future use. Overall, this structured approach provides a systematic and effective methodology for language classification tasks.

4.1 Data Collection and Exploration

4.1.1 Data Retrieval

The research begins by collecting the training and validation data for multiple languages, including Slovak (SK), Czech (cs), and English (en). The data files are read and stored in memory.

4.1.2 Data Statistics

To understand the characteristics of the dataset, descriptive statistics are computed. This includes information such as the number of sentences, words, unique words, and a sample extract from each language.

4.2 Data Cleaning and Preprocessing

4.2.1 Text Preprocessing

The collected data undergoes preprocessing to ensure uniformity and consistency. This involves:
Lowercasing the text. Replacing
hyphens with spaces.
Removing newline characters and punctuation marks.
Tokenizing sentences into words.

4.2.2 Preprocessed Data Statistics

The statistics of the preprocessed data are examined to observe any changes in the dataset after preprocessing.

4.3 Building the Multinomial Naive Bayes Model

4.3.1 Vectorizing Training Data

The text data is converted into numerical format using the CountVectorizer, which transforms the text into a bag-of-words representation. The outcome is a sparse matrix in which every row corresponds to a sentence, and each column represents a distinct word.

4.3.2 Initializing Model Parameters and Training

A Multinomial Naive Bayes model is initialized with default parameters, and it is trained onthe vectorized training data. The alpha parameter is used for smoothing to handle unseen words.

4.4 Vectorizing Validation Data and Model Evaluation

4.4.1 Vectorizing Validation Data

The validation data is preprocessed and vectorized using the same CountVectorizer that was fitted on the training data.

4.4.2 Model Evaluation

The validation data is utilized to make predictions using the trained Naive Bayes model. The model's performance is evaluated by employing the following metrics:

Confusion Matrix: To evaluate the accuracy of language classification.
F1 Score (Weighted): Providing an overall measure of model performance.

4.5 Fine-Tuning the Model

4.5.1 Model Adjustment

The model is fine-tuned by modifying the alpha parameter and setting fit_prior to False. This adjustment is made to improve the model's accuracy, especially when dealing with less common languages.

4.6 Using Sub words for Improved Model Perspective

4.6.1 Sub word Analysis

Subword analysis is performed to find common character sequences within words in the dataset. This involves:
Breaking words into characters.
Identifying common character sequences.
Converting dataset using these subwords.

4.6.2 Subword Preprocessing

The training and validation data are preprocessed to incorporate subword information, and the statistics of the subword-based dataset are examined.

4.7 Rebuilding and Evaluating the Model with Subwords

4.7.1 Vectorizing Subword Data

The data preprocessed with subwords is vectorized using the CountVectorizer, and the modelis rebuilt with this data.

4.7.2 Model Evaluation with Subword

The performance of the model using subword-based data is evaluated using the same metrics as before, including the confusion matrix and the weighted F1 score.

4.8 Model Persistence

The final trained Multinomial Naive Bayes model and the CountVectorizer are saved using job lib for future use.

5 Evaluation

Confusion Matrix:

The information pertaining to the accuracy of predictions for each language category (sk, cs, en) is presented, including true positive, true negative, false positive, and false negative predictions.

```
[[4886    0  114]
[4077  534  389]
[  0    0 5000]]
```

Here's a summary of the results:

Accuracy:

The model achieves an overall accuracy of approximately 84.56%. This means that about 84.56% of the sentences in the validation dataset are correctly classified into their respective languages.

Precision and Recall: The model's precision and recall values for each language category ('sk,' 'cs,' 'en') are not explicitly provided in the code output, but they can be calculated based on the confusion matrix.Precision measures the accuracy of positive predictions, while recall gauges the comprehensiveness of positive instances

F1-Score:

The model's overall performance can be evaluated using the weighted F1-Score, which is around 0.846. This metric considers the imbalance in class distribution and is computed as the harmonic average of preciseness and recall . It offers a balanced measure of the model's performance.

Misclassifications: The confusion matrix reveals that the model makes some misclassifications, as indicated by the off-diagonal elements. For example, some sentences in Slovak ('sk') are incorrectly classified as Czech ('cs') and vice versa. Similarly, there are misclassification between Czech and English. These misclassifications can be further investigated to improve the model's performance.

F1 Score (Weighted):

The F1 score assesses both precision and recall, making it a valuable performance measure of a model, providing a comprehensive assessment of its effectiveness. The "weighted" F1 score adjusts for classimbalance within the dataset.
F1 Score (Weighted): 0.6149824401040264 (Before adjustments)
F1 Score (Weighted): 0.8368507601649364 (After adjustments) F1
Score (Weighted): 0.8456381060126386 (With subwords)

6 Results

Model Stage	**Weighted F1 Score**
Initial Model (Before Adjustments)	0.615
Model Adjustments (Fine-Tuning)	0.837
Using Subwords for Improved Model	0.846

Table 1: F1 score of results in various stages

Initial Model (Before Adjustments):

The initial Multinomial Naive Bayes model achieved a weighted F1 score of approximately 0.615.
This score reflects the performance of the model on language classification before any adjustments.

Model Adjustments (Fine-Tuning):

After fine-tuning the model by modifying the alpha parameter and setting fit_prior to False,the model's weighted F1 score improved significantly to approximately 0.837.
These adjustments helped improve the model's accuracy, especially when dealing with less common languages.
Using Subwords for Improved Model Perspective:
By incorporating subword information into the dataset, you further improved the model's performance.
The model trained on subword-based data achieved a weighted F1 score of approximately 0.846.

The weighted F1 score suggests that incorporating subword analysis and fine-tuning model parameters has greatly enhanced the precision of your language detection model. As a result, the model is now more proficient in identifying Slovak, Czech, and English languages in text.In comparing the results of our Multinomial Naive Bayes model with existing methods(n-gram Models), it's evident that each stage of refinement has led to significant improvements in language classification accuracy. Initially, our model achieved a weighted F1 score of approximately 0.615, indicating moderate performance before any adjustments were made. However, through fine-tuning techniques, such as modifying the alpha parameter and setting fit prior to False, we observed a substantial enhancement in performance, with the weighted F1 score rising to approximately 0.837. These adjustments notably improved the model's accuracy, particularly in distinguishing less common languages. Furthermore, incorporating subword information into the dataset further boosted performance, resulting in a weighted F1 score of approximately 0.846. This enhancement suggests that the integration of subword analysis, along with fine-tuning model parameters, has significantly improved the precision of our language detection model. Consequently, our model now demonstrates enhanced proficiency in accurately identifying Slovak, Czech, and English languages within text data. This progression showcases the efficacy of our methodology in iteratively refining the model's performance and underscores its potential for advancing language classification tasks beyond traditional approaches.

7 Conclusion

The Multinomial Naive Bayes model demonstrates a commendable level of accuracy and performs reasonably well in classifying sentences into Slovak, Czech, and English languages. However, it is important to acknowledge that there are still some misclassifications occurring within the model's predictions. These misclassifications present opportunities for further improvement in both accuracy and precision.

Future work in this area could focus on several key areas to enhance model’s performing ability. Firstly, refining the feature selection process could lead to more accurate representations of language-specific characteristics, potentially reducing misclassifications. Additionally, exploring more advanced machine learning algorithms or ensemble techniques may offer improvements in classification accuracy by leveraging the strengths of different models.

Furthermore, incorporating contextual information and semantic analysis techniques could enhance the model's ability to discern subtle linguistic nuances, thereby reducing misclassifications, especially in cases where languages exhibit significant similarities. Moreover, expanding the dataset to include a more diverse range of texts and genres could help the model generalize better to various linguistic contexts and improve its overall performance.

Overall, while the Multinomial Naive Bayes model serves as a strong foundation for language classification, ongoing research and development efforts are essential to address remaining challenges and further refine its accuracy and precision. By embracing these future directions, we can continue to advance the effectiveness of language classification models and their applicability across a wide range of domains and industries.

8 References

1. Daniel W. Otter, Julian R. Medina and Jugal K. Kalita, "A Survey of the Usages of Deep Learning", Natural Language Processing, vol. 1, no. 1, pp. 35, July 2018.

2. ROBERT DALE, "The commercial NLP Landscape in 2017", Article in Natural Language Engineering, July 2017.

3. 3. ACL 2018: 56th Annual Meeting of Association for Computational Linguistics, [online] Available: https://acl2018.org.

4. Intelligent automation: Making cognitive real Knowledge Series I Chapter 2. 2018 EY report.

5. Jacques Bughin, Eric Hazan, Sree Ramaswamy, Michael Chui, Tera Allas, Peter Dahlström, et al., "MGI ARTIFICIAL INTELLIGENCE THE NEXT DIGITAL FRONTIER?", McKinsey & Company McKinsey & Company report July 2017, 2017.

6. Svetlana Sicular and Kenneth Brant, Hype Cycle for Artificial Intelligence 2018 Gartner report July 2018, 2018.

7. Oshin Agarwal, Funda Durupinar, Norman I. Badler and Ani Nenkova, "Word embeddings (also) encode human personality stereotypes", Proceedings of the Joint Conference on Lexical and Computational Semantics, pp. 205-211, 2019.

8. Silvia Quarteroni, "Natural Language Processing for Industry: ELCA's experience", Informatik-Spektrum, pp. 41, 2018.

9. Tom Young, Devamanyu Hazarika, Soujanya Poria and Erik Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]", IEEE Computational Intelligence Magazine, vol. 13, pp. 55-75, 2018.

10. Mohammad Hossein Amirhosseini, Hassan Kazemian, Karim Ouazzane and Chris Chandler, "Natural language processing approach to NLP meta model automation", International Joint Conference on Neural Networks (IJCNN), 2018, 8-13 July 2018.

11. Alan Ramponi and Barbara Plank, "Neural unsupervised domain adaptation in NLP—A survey", Proceedings of the 28th International Conference on Computational Linguistics, pp. 6838-6855, 2020.

12. Madhavi, K. Reddy, S. Viswanadha Raju, and J. Avanija. "Data Labeling and Concept Drift Detection using Rough Entropy For Clustering Categorical Attributes." HELIX 7, no. 5 (2017): 2077-2085.

13. Garrett Wilson and Diane J Cook, "A survey of unsupervised deep domain adaptation", ACM Transactions on Intelligent Systems and Technology (TIST), vol. 11, no. 5, pp. 1-46, 2020.

14. Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong and Quoc Le, "Unsupervised data augmentation for consistency training", Advances in Neural Information Processing Systems, pp. 33, 2020.

15. Su Lin Blodgett, Solon Barocas, Hal Daume and Hanna Wallach, "Language (technology) is power: A critical survey of "bias" in NLP", Proc. of ACL, 2020.