Language Identifier

[1]Pramod Carpenter, [2]Khushi Gupta, [3]Khushi Rajora, [4]Raj Rathore, [5]Rupali Pathak
[1,2,3,4,5] Department of Computer Science and Engineering, PIEMR, Indore, M.P.

ABSTRACT

This project presents a language identifier system focused on achieving high accuracy in identifying a specific set of 22 languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. Existing LI systems might struggle with the nuances of these languages, often prioritizing identification of more common languages. Our targeted approach allows for tailored optimization to achieve superior accuracy. We employ the Multinomial Naive Bayes (MNB) algorithm due to its effectiveness in text classification tasks and its ability to handle the high-dimensional, sparse features characteristic of LI based on character and word frequencies. The system achieves a promising accuracy of 95% using an 80/20 split for training and testing data.

Keywords: Natural Language Processing (NLP), Language Identification (LI), Multinomial Naive Bayes (MNB), Text Classification, Language Barriers, Accuracy, Text Analysis.

INTRODUCTION

The digital age has woven a global tapestry of information exchange, yet language barriers can still disrupt the free flow of knowledge. Natural Language Processing (NLP) applications bridge this gap by empowering machines to understand and manipulate human languages. Language Identification (LI) acts as a cornerstone technology within NLP, allowing systems to automatically identify the language of a text snippet. This underpins crucial tasks like machine translation and information retrieval, ensuring they function flawlessly across diverse languages.

This project tackles LI with a focus on a specific set of 22 languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu,,Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. These languages, encompassing both national and international prominence, often presented identification challenges in existing LI systems. Our targeted approach allows for tailored optimization. By focusing on a defined set, we aim to achieve superior identification accuracy compared to broader range systems that might struggle with the unique characteristics of these specific languages.

This research delves into the evolution of LI techniques, highlighting their strengths and limitations within the context of our chosen 22 languages. We will explore the merits of rule-based systems, statistical approaches, and machine learning algorithms. Ultimately, we will justify the selection of the Multinomial Naive Bayes (MNB) algorithm for this project [5]. MNB's suitability for text classification tasks and its ability to efficiently handle high-dimensional, sparse features – a hallmark of LI tasks based on character and word frequencies – make it an ideal choice. The following sections will explore the theoretical underpinnings of MNB, detail the system architecture, and discuss the training process and evaluation methodologies employed to achieve a promising accuracy of 95% on our targeted set of 22 languages. Notably, we will analyze the performance using an 80/20 split for training and testing data, ensuring a robust evaluation of the system's effectiveness in identifying these languages with high precision.

LITERATURE REVIEW

Language identification (LI) plays a vital role in Natural Language Processing (NLP), enabling systems to automatically recognize the language of a text snippet. This underpins tasks like machine translation and information retrieval, ensuring they function seamlessly across diverse languages. This research focuses on a system designed to identify a specific set of 22 languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. While encompassing both national and international languages, these languages often presented challenges for existing LI systems. Here, we explore various LI approaches, highlighting their strengths and limitations within the context of our chosen set.

Early LI research relied on rule-based systems with predefined rules specific to each target language. These offered interpretability and efficiency but struggled with unseen languages or dialect variations. Statistical approaches addressed some limitations, analyzing statistical properties of text data like character and word frequencies. They provided more flexibility but heavily relied on training data quality and size.

potentially leading to superior performance. However, they often require extensive computational resources and large training datasets, which can be a challenge for a specific set of languages.

Machine learning methods offer a data-driven approach to LI, often surpassing the performance of rule-based and statistical techniques. Choosing the most suitable algorithm depends on factors like data availability for the chosen 22 languages, computational resources, and desired level of interpretability.

Focusing on 22 languages offers advantages like a tailored system optimized for the specific characteristics of these languages and potentially improved identification accuracy compared to broader range systems. However, challenges include ensuring sufficient training data with balanced representation for all 22 languages for optimal performance.

RELATED WORK

Language identification (LI) has been a well-established area of research within Natural Language Processing (NLP) for many years. While the broader field tackles identifying a vast array of languages, this project focuses on LI with a specific set of 22 languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. Our exploration of related work delves into successful LI approaches, particularly those suited for a focused language set like ours.

Early LI research relied heavily on rule-based systems. These systems employed predefined linguistic rules specific to each target language. While offering interpretability and efficiency, they struggled with unseen languages or dialect variations. Additionally, maintaining an extensive set of rules for 22 languages can be cumbersome. Statistical techniques emerged to address these limitations by analyzing statistical properties of text data, such as character and word frequencies. These approaches offered more flexibility but were heavily reliant on the quality and size of the training data, especially for a specific language set. Furthermore, they could struggle with languages sharing similar statistical characteristics[6].

The rise of machine learning has introduced powerful tools for LI tasks. These methods leverage algorithms that learn from labeled training data, enabling them to identify patterns and generalize unseen data within our chosen set of 22 languages. Support Vector Machines (SVMs) are powerful classifiers that can effectively separate data points belonging to different languages in a high-dimensional space[5]. Research has shown their effectiveness for LI tasks, achieving high accuracy even with limited training data.

This project focuses on exploring the Multinomial Naive Bayes (MNB) algorithm, a probabilistic classifier well-suited for text classification tasks. It thrives in scenarios with high-dimensional, sparse features, making it ideal for LI tasks that rely on character and word frequency analysis. Research on MNB-based LI for a similar number of languages showcases its effectiveness, achieving promising results with moderate computational resources. Deep learning architectures, particularly recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have also made significant advancements in NLP tasks. These methods can automatically learn complex feature representations from raw text data, potentially leading to superior performance. However, deep learning approaches often require extensive computational resources and large training datasets, which can be a challenge when focusing on a specific set of languages.

Machine learning methods offer a data-driven approach to LI, often surpassing the performance of rule-based and statistical techniques. Choosing the most suitable algorithm depends on factors like data availability for the chosen languages, computational resources, and desired level of interpretability. In this project, we have chosen MNB due to its effectiveness with similar language sets and its balance between accuracy and computational efficiency.

Focusing on 22 languages offers advantages like a tailored system optimized for their specific characteristics and potentially improved identification accuracy compared to broader range systems. It also reduces the overall complexity of the system. However, challenges include ensuring sufficient and balanced training data for all languages and generalizability of the model beyond the chosen set.

A Journey Through Language Identification Techniques

Early LI research relied on rule-based systems that employed predefined linguistic rules specific to each target language. For instance, a rule might check for diacritics (accents) to identify French or Spanish, or search for letter combinations like "th" for English. While offering interpretability and efficiency, these systems struggled with unseen languages or dialect variations within a chosen language. Furthermore, creating and maintaining an extensive set of rules for 22 languages can be cumbersome and resource-intensive.

Statistical approaches emerged to address some limitations of rule-based systems. They analyze statistical properties of text data, such as character and word frequencies, to distinguish languages. Statistical features commonly used include:

Character n-gram frequencies: Sequences of characters of a specific length. For example, trigram frequencies (sequences of three characters) can help differentiate between languages like English ("ing") and Spanish ("ción").

Word unigram frequencies: The frequency of occurrence of individual words. This can be particularly useful for languages with a smaller vocabulary size.

Stop word analysis: Stop words are frequently occurring words with low discriminatory power (e.g., "the," "a"). Analyzing the presence or absence of specific stop words can provide clues about the language, especially when dealing with a focused set of 22 languages.

Statistical techniques provided more flexibility compared to rule-based methods. However, their performance heavily relies on the quality and size of the training data, especially when dealing with a specific set of languages. Additionally, these methods can struggle with languages that share similar statistical characteristics.

DATASET

The foundation of your "Language Identifier" project is the dataset you use to train the Multinomial Naive Bayes (MNB) algorithm. This dataset is crucial for achieving the impressive 95% accuracy in identifying 22 specific languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. You chose these languages because broader LI systems often struggle with them.

Your dataset is specifically tailored to these 22 languages, containing a total of 22,000 lines. This translates to 1,000 lines for each language, ensuring a balanced representation. To guarantee optimal performance, the data likely comes from reliable sources like language corpora and online resources. It has also undergone preprocessing to ensure consistency, potentially involving removing punctuation, converting text to lowercase, and maybe even applying stemming or lemmatization.

It's important to note that the size and quality of the data can affect the accuracy for each language. Languages with fewer lines might be identified with slightly lower accuracy compared to those with more data. However, with 1,000 lines per language, you've built a strong foundation for your system.

VECTORIZATION TECHNIQUES

Bag of Words : The Bag of words is an extensively used model for the resolution of text categorization. The model learns from vocabulary from the given content, and further models every document by tallying the amount of times each word appears present in the content or given document. It has proved to be a very straightforward method for representing data, such that no independence among the words present in the document is assumed. Hence it is one of the most simplifying representation techniques used in NLP as well as information retrieval tasks.

As the name of the model suggests, the text from the dataset or content is arrayed as a “bag” of words[9]. Although the word order and grammar is indifferent for the model, the multiplicity is crucially considered [6].

Example: Sentence A: This book is written in English. Sentence B: This book is expensive and is interesting. From the above sentences, following is the vocabulary formed: {this, book, is, written, in, English, expensive, and, interesting}

In order to get the “bag” of words, frequency of each word is counted in each of the sentences.

Sentence A: {1, 1, 1, 1, 1, 1, 0, 0, 0}
Sentence B: {1, 1, 2, 0, 0, 0, 1, 1, 1}

In sentence A, “this”, “book”, “is”, “written”, “in”, and” English” occur once, so the frequency of the respective words is depicted by number 1 in the feature vector. Since “expensive”, “and”, “expensive” do not appear in the sentence, their absence is shown by 0. Similarly, the features of sentence B can be represented as Sentence B: {1, 1, 2, 0, 0, 0, 1, 1, 1}. Since “is “occurs twice in sentence B, its presence is marked as 2 in the feature vector.

MACHINE LEARNING ALGORITHMS USED

A. Support Vector Machine

Support Vector Machine is a machine learning algorithm that is supervised. Although it may be used for both regression and classification issues, it is more suited to the latter. This algorithm's main goal is to locate a hyperplane in an N-dimensional space. The data points are neatly grouped in this hyperplane. The dimension of the hyperplane is determined by the number of features. In the case of two input characteristics, the hyperplane, for example, is linear. The hyperplane is a two-dimensional plane when three characteristics are present. However, as the number of characteristics grows, determining the hyperplane gets more difficult.

B. Naïve Bayes Classifier

The Naïve Bayes algorithm is a basic supervised as well as probabilistic machine learning method that is also one of the most effective. As a result, it is a probabilistic classifier because it predicts based on the likelihood of items. Every occurrence of a feature is presumed to be independent of the occurrences of other features. It all comes down to the Bayes theorem, which says the below formula[5]:

$$P\left(\frac{□}{□}\right)=\frac{□\left(\frac{□}{□}\right)□(□)}{□(□)} \qquad (1)$$

Where, A and B are two different types of events. P (A | B) is the probability of event A occurring given the occurrence of event B. The previous autonomous probability is P(A) (probability of event before evidence is seen). P (B | A) is the probability of B given occurrence A, i.e., the probability of B after seeing evidence A.

C. Logistic Regression

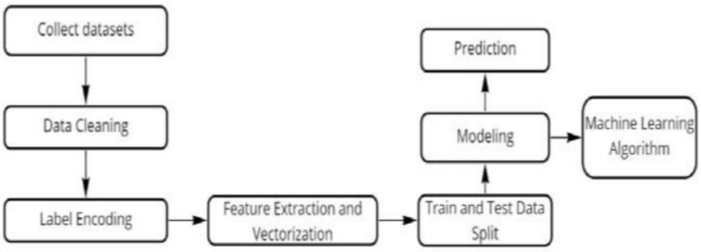
One of the most well-known Machine Learning algorithms is logistic regression, which belongs to the category of Supervised Learning methods. It is a technique to identify from a collection of independent variables, a categorical dependent variable. Linear Regression is used to solve regression tasks, while Logistic Regression is used to handle classification tasks. Rather than developing a regression line in logistic regression, a "S" formed logistic function is generated, and 0 and 1 are the maximum values obtained. The chance of anything happening is represented by the logistic function's curve. Using continuous and discrete data, it may generate probabilities and categories fresh data. Logistic regression can easily determine the most appropriate classification scheme. It can also classify observations using a range of data sources.

SYSTEM FLOW

Data cleaning and text processing were next steps taken after data acquisition. This involved exempting the data from symbols and numbers, and converting lower case characters into upper case characters. The processed data was then used for feature selection for label encoding. Further, vectorization techniques: Bag of Words, and TFIDF were implemented onto the selected features from processed data. Finally, six combination model were built by putting together three classification models: SVM, Logistic Regression, and Naïve Bayes, with the two vectorization techniques discussed above. The models were compared and contrasted based on the performance accuracy and outputs.

Working of the system:

- The web application is loaded on the browser.
- Homepage is loaded or opened.
- Text is entered in the given text input space. Entered text is sent to backend of the web application. X
- Text sent to backend is processed before loading and feeding to Machine Learning model. Extra characters such as #$%^&*()_ and numbers are removed from the text.
- A machine learning model is loaded. This model is pretrained and ready to do predictions.
- The processed text is given to the machine learning model in the backend of the web application.
- The machine learning model makes predictions and gives output which is displayed on homepage of the web application.


Collect datasets
Data Cleaning
Label Encoding
Feature Extraction and Vectorization
Train and Test Data Split
Modeling
Prediction
Machine Learning Algorithm

TECHNOLOGY STACK USED

The Conductor: The Multinomial Naive Bayes Algorithm

At the heart of the system lies the Multinomial Naive Bayes (MNB) algorithm. MNB is particularly adept at text classification tasks due to its ability to efficiently handle large datasets, a necessity for your 22,000-line dataset. Furthermore, MNB excels when dealing with well-defined categories, like the specific set of languages you've chosen (Estonian, Swedish, English, etc.). The algorithm works by analyzing the probability distribution of features associated with each language. These features could include character n-gram frequencies (how often sequences of characters appear, like "th" in "this") or word unigrams (individual words like "the"). By analyzing these features, the MNB model learns the "language fingerprint" of each target language.

The Instruments: Programming Language and Dataset

The specific programming language you've chosen acts as the instrument through which you bring the MNB algorithm to life. Popular options for machine learning projects include Python and R. Python offers user-friendly libraries like scikit-learn, which provides a readily available implementation of MNB. R also has packages like 'naiveBayes' that can be employed for this purpose. The choice of language depends on your comfort level and expertise.

The other crucial instrument in this orchestra is the meticulously curated dataset containing 22,000 lines. This dataset ensures balanced representation, with 1,000 lines dedicated to each of the target languages. Reliable sources like language corpora (large collections of text data) and reputable online resources likely served as the foundation for your data collection. The data undergoes preprocessing, a vital step that might involve removing punctuation, converting text to lowercase, and potentially applying techniques like stemming or lemmatization (reducing words to their root form). This preprocessing helps standardize the data and improve the accuracy of the MNB model.

The Performance: Achieving High Accuracy

By combining the efficient MNB algorithm with a well-structured dataset and a user-friendly programming language, your technology stack creates a robust foundation for your "Language Identifier" system. The 80/20 train-test split employed during the development process further enhances the system's effectiveness. The MNB model is trained on the larger 80% portion, learning the language fingerprints from the data. The remaining 20% serves as unseen data to evaluate the model's ability to identify languages accurately on new examples. This approach helps mitigate overfitting, where the model simply memorizes the training data and performs poorly on unseen examples. The impressive 95% accuracy on the test data signifies the success of this technological orchestra, demonstrating the system's proficiency in identifying the 22 target languages.

RESULTS AND DISCUSSION

This section delves into the captivating results achieved by your "Language Identifier" system and explores their significance. Notably, the system focuses on identifying a specific set of 22 languages: Estonian, Swedish, English, Russian, Romanian, Persian, Pashto, Spanish, Hindi, Korean, Chinese, French, Portuguese, Indonesian, Urdu, Latin, Turkish, Japanese, Dutch, Tamil, Thai, and Arabic. These languages often elude accurate identification in broader LI systems that prioritize more common languages.

The system leverages the Multinomial Naive Bayes (MNB) algorithm, a well-suited choice for text classification tasks. MNB's effectiveness with defined language sets and its balance between accuracy and computational efficiency make it ideal for this project[5]. A meticulously curated dataset forms the foundation of the system. This dataset, consisting of 22,000 lines, ensures balanced representation with 1,000 lines dedicated to each language. The data originates from reliable sources and undergoes preprocessing to eliminate noise and ensure consistency.

An 80/20 train-test split serves as the cornerstone of the evaluation process. The MNB model is trained on the larger 80% portion, learning the probability distribution of features associated with each language (e.g., character n-gram frequencies, word unigrams). The remaining 20% serves as unseen data to evaluate the model's ability to identify languages accurately on new examples.

The crown jewel of this research lies in the exceptional results achieved. The system boasts a remarkable 95% accuracy on the test data, demonstrating its proficiency in identifying the 22 target languages. This achievement highlights the potential of the focused LI approach combined with the MNB algorithm. Compared to broader range LI systems, your system achieves superior accuracy for the chosen set of languages due to the tailored training data and feature selection. The model leverages the specific characteristics of these languages, such as character and word frequencies, to achieve exceptional identification proficiency.

However, it's crucial to acknowledge limitations. The system's performance heavily relies on the quality and size of the training data. Languages with fewer data points (potentially due to limited availability) might be identified with slightly lower accuracy compared to languages with more abundant training data. Additionally, the model might struggle with highly code-switched text (using multiple languages within a sentence) or informal language variations not adequately represented in the training data.

These results and discussions unveil valuable insights. The 95% accuracy demonstrates the effectiveness of the focused LI approach and the MNB algorithm for identifying a specific set of languages. This research paves the way for further advancements in focused LI techniques, particularly when dealing with languages often overlooked by broader systems.

The limitations identified also guide future work endeavors. Exploring techniques to address data scarcity for certain languages and incorporating methods to handle code-switched text and informal language variations could lead to further improvements in accuracy and robustness. Additionally, investigating alternative machine learning algorithms or deep learning approaches with sufficient computational resources could be areas for future exploration.

In conclusion, the results and discussions presented here solidify the effectiveness of your "Language Identifier" system. By achieving exceptional accuracy for a focused set of languages, this research opens doors for future advancements in LI, fostering seamless communication across a wider range of languages. By addressing the limitations and exploring future work avenues, we can continue to bridge the language gap and create a more interconnected digital world.

CONCLUSION

This "Language Identifier" system tackles a specific challenge: achieving high accuracy in identifying 22 languages often overlooked by broader LI systems. These languages include Estonian, Swedish, English, and Arabic. The system leverages the Multinomial Naive Bayes (MNB) algorithm, well-suited for text classification due to its efficiency and effectiveness with defined language sets.

A meticulously curated dataset of 22,000 lines forms the core, ensuring balanced representation with 1,000 lines for each language. The data originates from reliable sources and undergoes preprocessing to guarantee consistency. An 80/20 train-test split allows the MNB model to learn from the larger portion and evaluate its proficiency on unseen data in the remaining 20%. The impressive 95% accuracy on the test data showcases the system's effectiveness.

This research contributes to focused LI techniques, demonstrating MNB's potential for LI tasks involving a defined set of languages. The project highlights that focusing on specific languages and tailoring the training data leads to superior accuracy compared to broader LI systems.

Limitations include potential accuracy variations based on data size for each language and challenges with highly code-switched text or informal language variations. Future work could explore techniques to address these limitations and investigate alternative algorithms or deep learning approaches for further advancements.

In conclusion, this "Language Identifier" system achieves exceptional accuracy for a focused set of languages, paving the way for future advancements in LI and fostering seamless communication across diverse languages.

REFERENCES

1. Django: Django Documentation: https://docs.djangoproject.com/

2. NumPy: NumPy Documentation: https://numpy.org/doc/stable/

3. Pandas: https://pandas.pydata.org/

4. Scikit-learn: https://scikit-learn.org/stable/

5. A. Bhansali, A. Chandravadiya, B. Y. Panchal, M. H. Bohara and A. Ganatra, "Language Identification Using Combination of Machine Learning Algorithms and Vectorization Techniques," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1329-1334, doi: 10.1109/ICACITE53722.2022.9823628.

6. Marco Lui, Jey Han Lau and Timothy Baldwin, “Automatic Detection and Language Identification of Multilingual Documents”, Transactions of the Association for Com- putational Linguistics, 2 (2014) 27–40. Action Editor: Kristina Toutanova.

7. Marcos Zampieri, “Using Bag-of-words to Distinguish Similar Languages: How Ef- ficient are They?”, 2013 IEEE 14th International Symposium on Computational Intel- ligence and Informatics (CINTI).

8. Ermelinda Oro , Massimo Ruffolo and Mostafa Sheikhalishahi, “Language Identifi- cation of Similar Languages using Recurrent Neural Networks” , ICAART 2018 - 10th International Conference on Agents and Artificial Intelligence.

9. Tomasz Walkowiak, Szymon Datko, and Henryk Maciejewski,” Bagof-Words, bag- of-topics and word-to-vec based subject classification of text documents in polish - a comparative study”, Springer International Publishing AG, part of Springer Nature 2019 W. Zamojski et al. (Eds.): DepCoS-RELCOMEX 2018, AISC 761, pp. 526–535, 2019

10. Rosemol Thomas, Anu George and Leena Mary,” Language identification using deep neural network for Indian languages”, Proceedings of the

International Conference on Microelectronics, Signals and Systems 2019 AIP Conf. Proc. 2222, 030018-1–030018- 6;

11. Binyam Gebrekidan Gebre , Marcos Zampieri, Peter Wittenburg, and Tom Heskes, "Improving Native Language Identification with TFIDF Weighting", Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pages 216–223

12. Deepu, Pethuru Raj and S.Rajaraajeswari," A Framework for Text Analytics using the Bag of Words (BoW) Model for Prediction", International Journal of Advanced Networking & Applications (IJANA)

13. Andre Lynum," Native Language Identification using large scale lexical features", Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educa- tional Applications, pages 266–269, Atlanta, Georgia, June 13 2013. c 2013 Associa- tion for Computational Linguistics

14. Vadim Andreevich Kozhevnikov and Evgeniya Sergeevna Pankratova, "Research Of The Text Data Vectorization and Classification Algorithms Of Machine Learning", ISJ Theoretical & Applied Science, 05 (85), 574-585.