e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Conference:01/December-2022 Impact Factor- 6.752 www.irjmets.com
1st National Conference on Applications of soft Computing Techniques in Engineering NCASCTE-2022
Organized by Department of Electrical & Electronics Engineering, ACE Engineering College, Hyderabad

LANGUAGE DETECTION USING MACHINE LEARNING

K Prem Kumar *1, T SRI VINAY*2, S Sai Aasritha*3, P Vasantha*4

*1Associate Professor, Department of Computer Science and Engineering ,ACE Engineering College, Ghatkesar, Telangana,India

*2,*3,*4Btech Student,Department of Computer Science and Engineering ,ACE Engineering College, Ghatkesar, Telangana,India

DOI: https://www.doi.org/10.56726/IRJMETS-NCASCTE202219

ABSTRACT

In today's globalized world, where people from all over the world are able to communicate and share information with each other. Language detection is important as it helps to bridge the gap between different cultures and languages. Language detection is a useful tool in many applications, such as machine translation, text summarization, and sentiment analysis. In these applications, it is important to first determine the language of the text before processing it further. Overall, language detection using NLP is a powerful tool for analyzing and processing natural language data. It can be used in a variety of applications to improve the accuracy and efficiency of natural language processing systems.

Keywords--Naive Bayes, Language Detection, Machine Learning, Natural Language Processing.

I. INTRODUCTION

Language identification can be a crucial step in a Natural Language Processing (NLP) problem. It will try to identify the natural language from a text. Language identification of a text is very important before language translation, sentiment analysis can be taken. For example, when we want to translate a language in google translate the box you type in says ‘Detect Language. The reason is firstly Google translate is trying to identify the language of the sentence which we gave before translating it. Language identification can be implemented with the help of Neural Network, character n-grams, Frequent word-based approach etc. Many researchers have developed the models that can predict different languages with different algorithms. Research on language identification began in the 1970’s.

At the end of almost 5 decades of research, we have seen that language recognition has been practiced in different ways. Many attempts by different researchers have been made to achieve maximum efficiency

II. METHODOLOGY

Previous Work

In “**n-gram and Decision Tree-based Language Detection for written words**” By JuhaHakkinen and Jilei Tian, N- Gram-Based approach is used. Intuitively, common words such as determiners, conjunctions and prepositions are good clues for identifying the language.

The n-gram method uses letter n-grams, which represents the frequency of occurrence of different n-letter combinations in a particular language. Language identification process can be divided into two phases: Training and Identification. A language identification model is trained for each targeted language. In the training phase, a list of words with a known language are presented as alphabetic strings. The frequency of occurrence of sequence of consecutive n letters is estimated from a large language specific training sample. Since it is not feasible to train all the possible partial letter sequence probabilities, a simplifying assumption is made that the probability of the current word depends only on the previous n-1 letters, which can be implemented using n-grams.

In Decision Trees Based Approach, Decision trees are used to determine the most likely language for each letter in word. The language is obtained by asking a series of questions about the context of the current letter, as defined by the corresponding decision tree. Since only the letter context is used and no frequency information is stored in the tree, a very compact representation is obtained. During training, the decision tree is grown by splitting the nodes into child nodes. Language tags are first generated for each letter of the input word. The decision tree corresponding to the letter in question is selected. The tree is climbed starting at the root node, by answering the questions presented by the attributes, until a leaf is found, or no answer to the question is found.

In "**Language Identification from Text Documents**" by Priyank Mathur, Arkajyoti Misra, Emrah Budur, Recurrent Neural Networks are used. RNNs are a kind of neural networks which possess an internal state by virtue of a cycle in their hidden units. As such, RNNs are able to record temporal dependencies among the input sequence, as opposed to most other machine learning algorithms where the inputs are considered independent of each other. Neural networks are also known for their natural language processing tasks and have been successfully used for applications like hand writing recognition, speech recognition etc.

The language detection service is used to identify the language of business texts, such as emails and chats. The service identifies the language of a text and the parts of that text where the language changes, down to the word level. Using the language detection service, Surveillance Insights can highlight and annotate the languages that are used in a text and help to identify potential suspicious activity.

Most of the existing systems use Neural networks, Support vector machines, Decision tress and other complex Machine Learning algorithms. Drawbacks of neural networks like Hardware dependent, black box in nature and data dependency are the factors should be considered while designing a learning model. If we want to train the model with more languages we have to take care of those factors. Training neural networks needs a lot of training data, often much more than that required for regular machine learning algorithms. Coming to decision sometimes calculation can go far more complex compared to other algorithms, Decision tree often involves higher time to train the model.

However, most of the algorithms need large datasets for training the learning model, also needed high processing power systems to train the model. More time is required to train the learning model.

Multinomial Naive Bayes Classifier

Our Proposed system is implemented with the help of Multinomial Naive Bayes theorem.A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The most important point of the classifier is based on the Bayes theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 1: Bayes Theorem

This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document. which does not require large datasets for training and testing. It does not need more time for training the model. Using Naive Bayes text classifiers might be a clever idea, especially if there's not much training data available and computational resources are scarce. Results are competitive in terms of performance if features are well engineered. Also, recent research concluded that Naive Bayes is one of the best algorithm in text classification.

Naive Bayes takes very less training time, compared to other algorithms, and give results which is almost competitive with other algorithms. As we also creating web interface using python flask, it will be user friendly for identifying languages. There are few websites in internet which can predict the languages, but the accuracy is extremely low, and they are predicting incorrect languages.

III. MODELING AND ANALYSIS

The data for this work was obtained from Wikipedia and other sources with the help of python using web scraping. The Data set contains different languages along with the sample text.

Languages like English, Russian, Tamil, Hindi etc were also included in this data set. The csv file contains language and text fields.

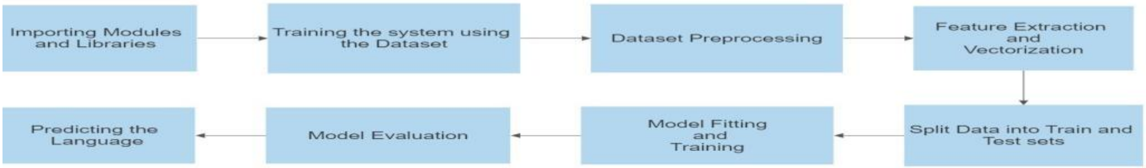
Importing Modules and Libraries
Training the system using the Dataset
Dataset Preprocessing
Feature Extraction and Vectorization
Split Data into Train and Test sets
Model Fitting and Training
Model Evaluation
Predicting the Language

Figure 2: Architectural Design

Step 1 : The first step is to import the all required libraries. Libraries used in this work are:

1. pandas
2. numpy
3. Matplotlib
4. Sklearn etc.

Step 2 : Now we have to separate the independent and dependent features. Here independent feature is text and dependent feature is language.

Step 3 : After separating the dependent and independent variable, Label encoding is performed. As our output, language is a categorical variable, we need to convert it into a numerical form,

Step 4 : The most important step is text pre-processing because results depends on how well data has been processed. All the unwanted symbols,numbers and punctuation marks will be removed in this phase.

Step 5 : Computer can only understand the data which is in numerical form, but our input feature is not in numerical form. So Bag of words is used to convert the input feature into numerical form.

Step 6 : Now the training and testing data should be splitted. Normally 80% of the data is used for training the model and remaining 20% of the data is used for testing the model.

Step 7 : Now we have to train the model by using the training data. After training we have to test the data with the testing data.

Step 8 : After training and testing, we have to evaluate the model.

Step 9 : Test with some more data.

Step 10 : Save the model.

IV. RESULTS AND DISCUSSION

After training the model with Multinomial Naive Bayes classifier , accuracy score of 0.974 is obtained. Even with the less training data and less training time, the model obtained an accuracy score which is almost competitive with the algorithms like Neural Networks, Decision Trees etc.

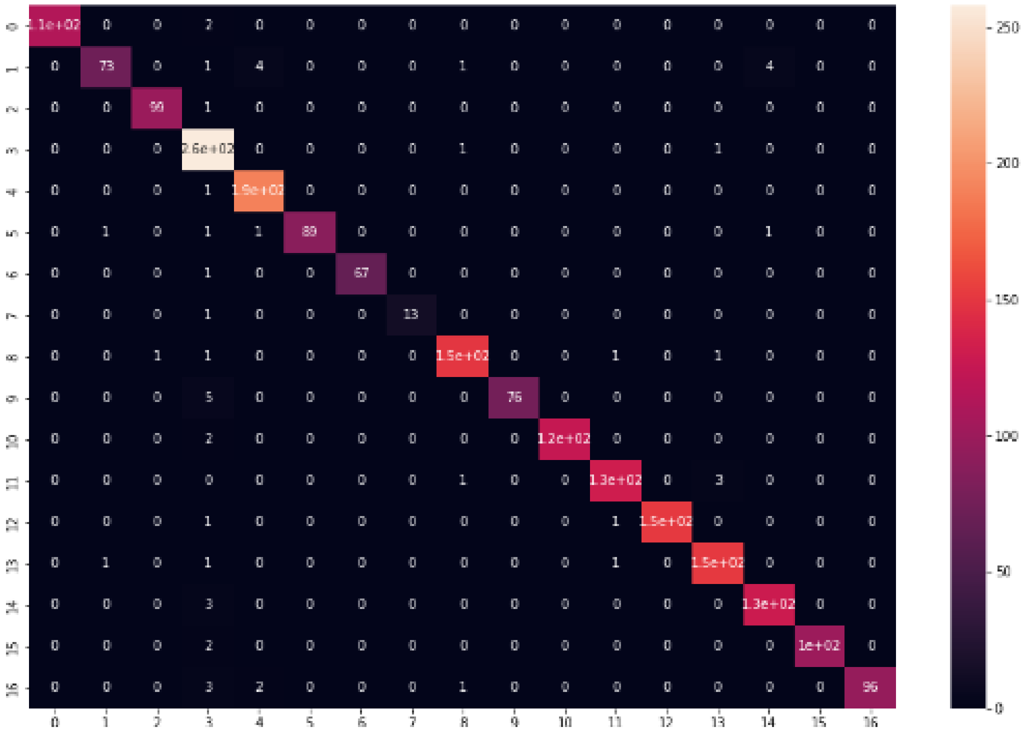

Figure 3: Heatmap

V. CONCLUSION

Language Detection is an important process in NLP applications. However, different algorithms are proposed to detect a language. Algorithms like Neural networks, Decision Tress, Support Vector Machines etc are used. In this paper we implemented language detection using Naive Bayes , which is one of the best text classification algorithms. If we want to detect any language we need internet connectivity, Advantage of our implemented system is it doesn't require any internet connectivity.

VI. REFERENCES

[1] Juha Hakkinen, jilei tian, "n-gram and Decision Tree based Language Detection for written words".

[2] Priynak Mathur, Arkajyoti Misra, Emrah Budur, "Language Identification From Text Documents".

[3] Belainine Billal, Alexsandro Fonseca, FatihaSadat, "Efficient natural Language Preprocessing for analyzing large datasets".

[4] Feng-Jen Yang, "An implementation of Naive bayes Classifier".

[5] Wisam A.Qader, Musa M.Ameen, Bilal I.Ahmed" An overview of Bag of Words Importance, Implementation, Applications and Challenges".