Language detection using NLP

Disha Shetiya, Pranita Davankar

Department of Computer Science Engineering

MIT ADT, Loni Kalbhor, Pune.

Guide: Prof. Gurunath Waghale

*Abstract:* We believe that Natural Language Processing is important because it helps in resolving ambiguity in language and adds useful structure to data for many down-stream applications such as speech recognition or text analysis. This system applies applications of Natural Language Processing to identify dominant language of document in the format of text and speech. It makes analysis process much easier. In this work, proposed system is able to predict the language of given text or given voice as a output. This is a solution for many artificial intelligence applications and computational linguists. These kinds of prediction systems are widely used in electronic devices such as mobiles, laptops, etc. for machine translation, and also on robots. It helps in tracking and identifying multilingual documents too. The domain of NLP is still a lively area of researchers.

I *Index Terms:* Speech recognition, Multilingual documents, Text analysis.

II. INTRODUCTION

Communication and interaction are much needed to stay connected with the world and to walk along with the world. We focus on these aspects as a social entity. If a speaker is taken to mean what he/she wants to be taken to mean, then the communication will be successful and effective. The project ‘Language Identifier’ helps for multilingual

language speaking people to avoid the barrier of interaction and improving analysis.

If we take scenario of email, some email might fool many languages identifier into tagging the email as English. This system helps to detect the regions with each document by correctly tagging the language of each section. To improve the categorization of short text and improve the search results, we can use Language Identifier. This helps to identify in which language tweets or comments have been posted on social media and make it easy to analyze them. Language identifier also helps in multilingual document to identify dominant language of content being used in body of text and in other sections too.

III.EXISTING SYSTEM

Automatic Speech Recognition system is there but can be useful in other domains (e.g., language identification). As an example, use case for Common Voice, this project present speech recognition experiments using Mozilla’s Deep Speech Speech-to-Text toolkit. For most of the languages, these are the first ever published results on end-to-end Automatic Speech Recognition. This project represents the speech recognition over the voice module. The limitation of this project was, it does not identify the language or count of words. It just recognizes the multilingual speech and convert it into text format but can be useful in language identification process.

There is also a system present which uses deep convolution neural network. The system addresses the problem of language identification from a computer vision perspective. System extracts the target language of a given audio sample by utilizing a hybrid network constructed of a Convolutional Neural Network (CNN) combined with a Recurrent Neural Network (RNN). In this work, system utilizes the power of CNNs to capture spatial information, and the power of RNNs to capture information through a sequence of time steps for identifying the language from a given audio snippet. They developed a DNN based on a sequence recognition network presented by Shi et al. In this section, they present the datasets they used for training the network, the audio representation used for training their models, and the structure of their proposed network in detail. This project lack in diversity of languages because of lack of dataset. This system expects new applicability of this model in various scenarios and in diverse of language.

IV.OBJECTIVE

To detect the language over text.

To detect multilingual languages.

V. TECHNOLOGY

V.I DATASET

Since there are no large-scale, freely available datasets for LID tasks (datasets such as the NIST Language Recognition Evaluation are only available behind a paywall), we resorted to creating our own datasets for our experiments. We collected our datasets from two different sources: (1) We used dataset from Kaggle to process our input for our language detection system (2) we sourced data from news broadcast channels hosted on YouTube. We chose to collect data for 6 different languages, while making sure that we include languages with similar phonetics. Following this idea, we collected data for English, German, French, Spanish, Russian, Mandarin, Chinese. Each audio clip is recorded in the speaker’s native language and features only one speaker. The dataset consists of many different female and male speakers. From this dataset we collected hours of speech data in four languages: English, German, French and Spanish. YouTube News Collection We chose to use news broadcasts as a second data source to obtain audio snippets of similar quality to the EU Speech Repository (different speakers, mostly one speaker at a time and a single defined language). We gathered all data from YouTube channels. The obtained audio data has many desired properties. The quality of the audio recordings is very high and hundreds of hours are available online. News programs often feature guests or remote correspondents resulting in a good mix of different speakers.

V.II WORKING

Separating Independent and Dependent features

Now we can separate the dependent and independent variables, here text data is the independent variable and the language name is the dependent variable. X = data["Text"] y = data["Language"]

Label Encoding

Our output variable, the name of languages is a categorical variable. For training the model we should have to convert it into a numerical form, so we are performing label encoding on that output variable. For this process, we are importing LabelEncoder from sklearn. from sklearn. preprocessing import LabelEncoder le = LabelEncoder () y = le.fit\_transform(y)

Text Preprocessing

This is a dataset created using scraping the Wikipedia, so it contains many unwanted symbols, numbers which will affect the quality of our model. So we should perform text preprocessing techniques.

# creating a list for appending the preprocessed text data\_list = [ ]

# iterating through all the text for text in X:

# removing the symbols and numbers text = re.sub(r'[!@#$(),n"%^\*?:;~`0-9]', ' ', text) text = re.sub(r'[[]]', ' ', text) # converting the text to lower case text = text.lower() # appending to data\_list data\_list.append(text)

Bag of Words

As we all know that, not only the output feature but also the input feature should be of the numerical form. So we are converting text into numerical form by creating a Bag of Words model using CountVectorizer. from sklearn.feature\_extraction.text import CountVectorizer cv = CountVectorizer () X = cv.fit\_transform(data\_list).toarray() X.shape # (10337, 39419)

Train Test Splitting

We preprocessed our input and output variable. The next step is to create the training set, for training the model and test set, for evaluating the test set. For this process, we are using a train test split. from sklearn.model\_selection import train\_test\_split x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20)

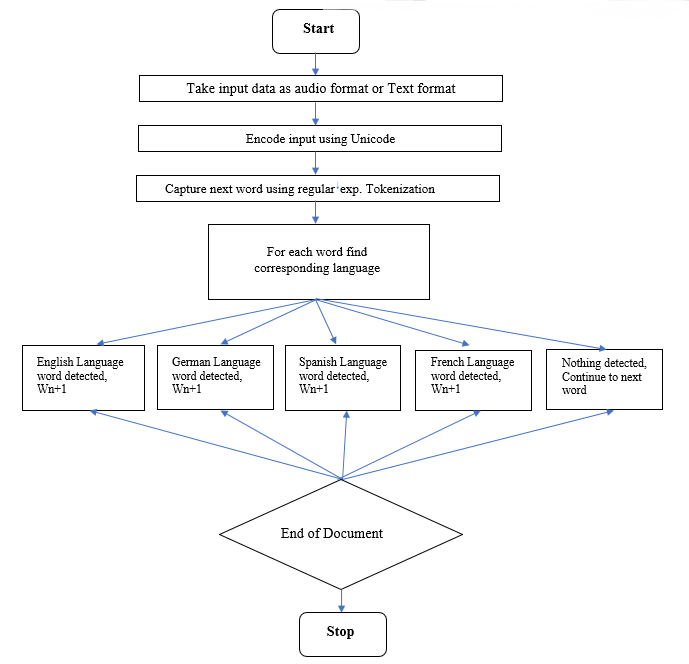
Model Training and Prediction

And we almost there, the model creation part. We are using the naive\_bayes algorithm for our model creation. Later we are training the model using the training set. from sklearn.naive\_bayes import MultinomialNB model = MultinomialNB() model.fit(x\_train, y\_train) So we’ve trained our model using the training set. Now let’s predict the output for the test set. y\_pred = model.predict(x\_test)

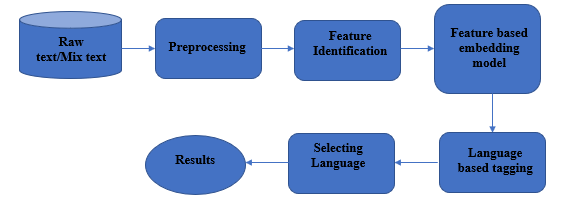
Model Evaluation

Now we can evaluate our model from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report ac = accuracy\_score(y\_test, y\_pred) cm = confusion\_matrix(y\_test, y\_pred) print("Accuracy is :",ac) # Accuracy is : 0.9772727272727273 The accuracy of the model is 0.97 which is very good and our model is performing well. Now let’s plot the confusion matrix using the seaborn heatmap. plt.figure(figsize=(15,10)) sns.heatmap(cm, annot = True) plt.show().

VI. FLOWCHART



VII. BLOCK DIAGRAM



VIII. RESULTS

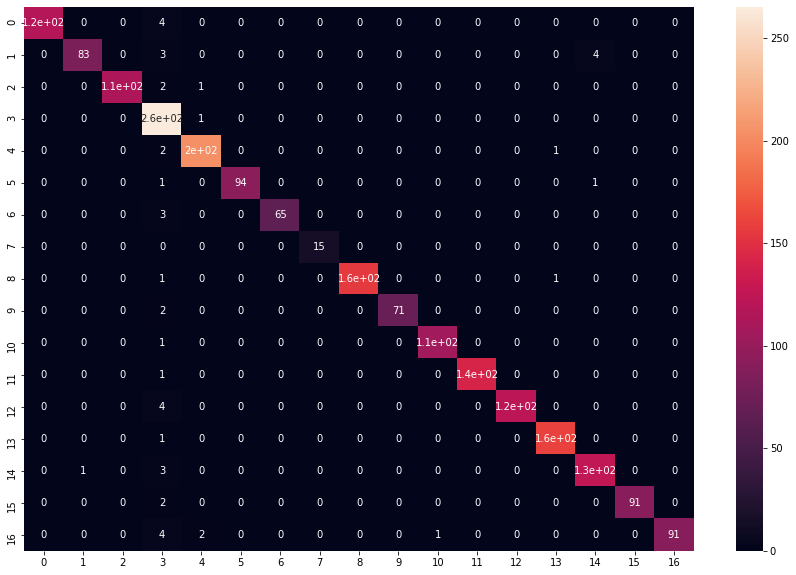


Fig. Graph

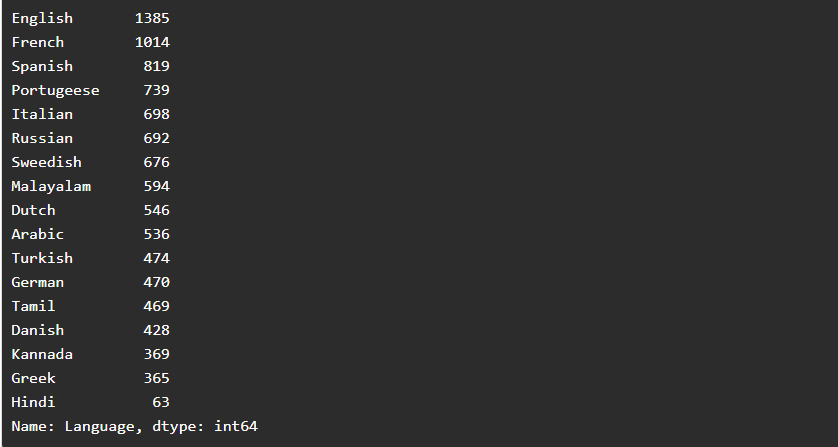


Fig. Word Count

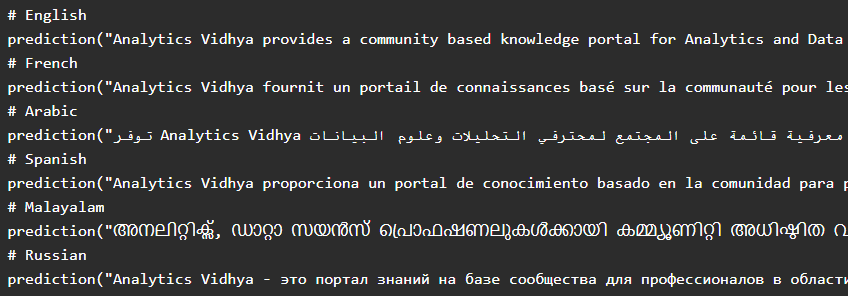


Fig. Detection of language

IX. FUTURE WORK

The results obtained so far clearly show that natural language methods of language identification are definitely worth considering in the scope of software language reverse engineering. In the future we plan to run more convincing experiments positioning best NLP-based methods among other approaches to SLI, highlighting corner cases of pairs of software languages commonly confused by one family of methods but not by the other.

We also expect certain types of software (e.g., compiler sources) to confuse many methods. False positives and false negatives should be inspected manually to determine causes for misclassification, possibly followed by recalibration of the chosen classifier.

X. CONCLUSION

We have worked on how to use different aspects of natural language processing and implement it in the project of language identification. We need to analyze the data and preprocess it accordingly. A bag of words model becomes a way of representing your text data. Text extraction and vectorization are important steps for good predictions in NLP. Naive Bayes always proves to be a better model in such text classification problems, hence more accurate results we get. This project would help in exploring various perspectives of new technology.

XI. References

[1] Language Detection using NLP

[Basil Saji](https://www.analyticsvidhya.com/blog/author/basilb2s/) — March 12, 2021

<https://www.analyticsvidhya.com/blog/2021/03/language-detection-using-natural-language-processing/>

# [2] 4 NLP Libraries for Automatic Language Identification of Text Data in Python

# Zoumana Keita

<https://towardsdatascience.com/4-nlp-libraries-for-automatic-language-identification-of-text-data-in-python-cbc6bf664774>

[3] Language Identification from Written Text- An Application of NLP

Shivani Sharma

Aug 13, 202

<https://medium.com/@btbtc18049_shivani/language-identification-from-written-text-an-application-of-nlp-eddf0caefa7d>.

[4] An Anatomization of Language Detection and Translation using NLP Techniques.

Bhagyashree Pujeri, Jagdeeshsai D

December 2020

<https://www.researchgate.net/publication/349398196_An_Anatomization_of_Language_Detection_and_Translation_using_NLP_Techniques>.

[5] Improved text language identification for the South African languages

Bernardt Duvenhage, Mfundo Ntini, Phala Ramonyai

18 January 2018

<https://ieeexplore.ieee.org/abstract/document/8261150>.

XII. BIBLIOGRAPHY

1. NLP – Natural Language Processing
2. Multilingual- Multiple Languages.
3. CNN - Convolutional Neural Network
4. CV – Count vectorizer.