Question type Classification

Jishnu Sai Sukesh
Siva Prasad Reddy
Dharani Priyanka
Priyanka Betha

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITETET

Photo: Ronny Østnes / OsloMet

Table of contents

1. Introduction

2. Methodology

3. Corpus Description

4. Statistical Methods

5. Model Implementation

6. Experiments and Results

7. Future Work and Conclusions

8. References

Introduction

- The objective of this project is to build a supervised learning model that can receive a question as input and detects and classify the given question based on it's domain using predefined set of domain category.
- The following are the eight category labels that are present in the predefined set of domain categories such as PERSON, NUMBER, PLACE, DATE, LOCATION, MONEY, ORGANISATION, PERCENTAGE.
- We used the following to develop a model in this project-
  Classifiers we used: Logistic Regression, Support Vector Machine, Multi-Layer Perceptron, Naive Bayes, Random Forest.
  Neural Networks models such as CNN, RNN-LSTM, RNN-GRU, Bidirectional RNN. models.

Methodology

- **Dataset Preparation**: The first step is the Dataset Preparation step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then splitted into train and validation sets.
- **Feature Engineering**: The next step is Feature Engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model.

  This step also includes the process of creating new features from the existing data.
- **Model Training**: The final step is the Model Building step in which a machine learning model is trained on a labelled dataset.
- **Improve Performance of Classifier**: we can improve the performance by using various models.

Corpus Description

- In NLP, **corpus** refers to a collection of texts may be formed of a single language of texts, or can span multiple languages.
- **Statistics:** The values for our data analysis are as follows:

  NUMB(Number)- 438

  PERS(Person)- 321

  LOCA(Location)- 315

  DATE(Date)- 136

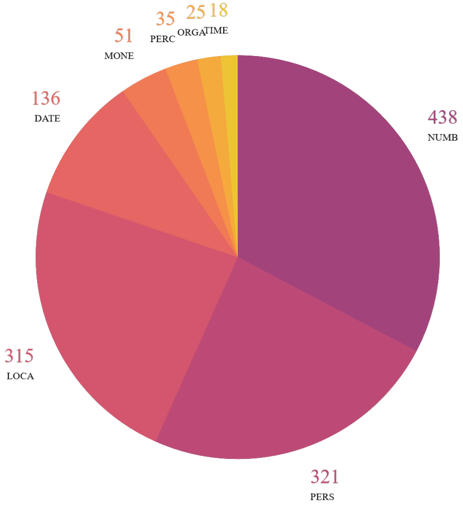
  MONE(Money)- 51

  PERC(Percentage)- 35

  ORGA(Organisation)- 25

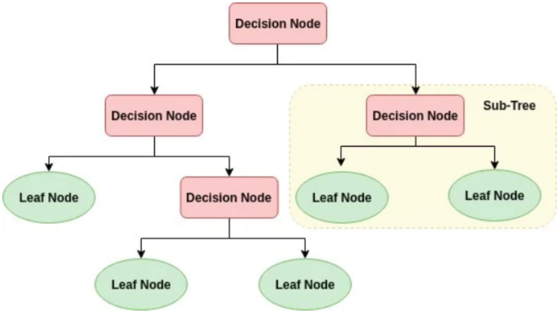
  TIME(Time)- 18

- **Plot:** Pie chart representing the above statistics of Corpus

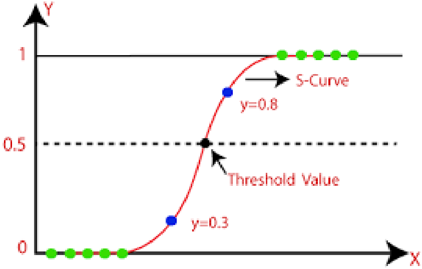


Statistical Methods

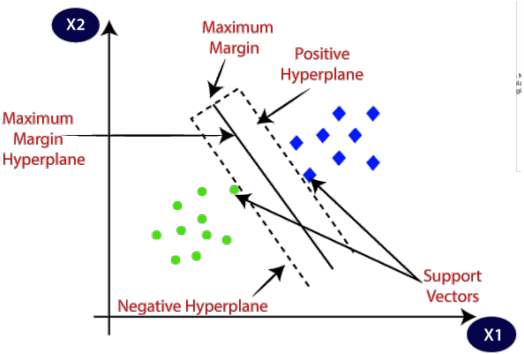
We used the following Statistical models to develop this project:
Decision Tree: Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



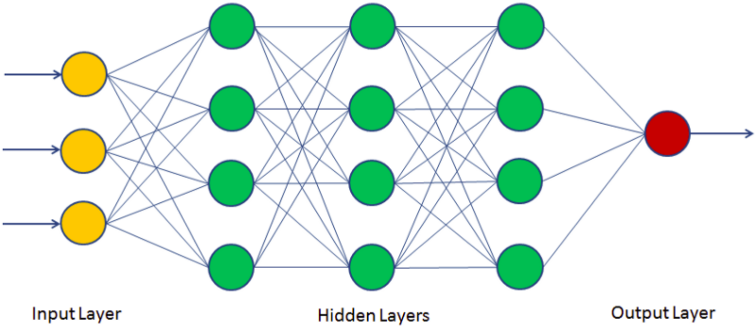
1. Logistic Regression(LR): Logistic Regression is a regression analysis which is conducted when the dependent variable is binary. It classifies the data into 0 or 1. It uses sigmoid function whose shape looks like ’S’to plot the data into a graph. Text classification is done based on the probabilities which lies between 0 and 1. We use Train-test-split method to train and test the data.



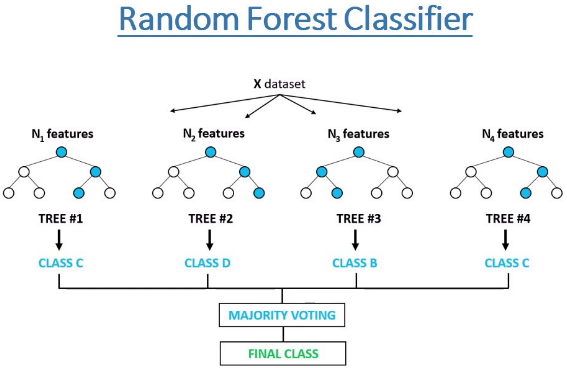
2. Support Vector Machine(SVM): The objective of the Support Vector Machine algorithm is to find a hyperplane that distinctly classifies the data points. Hyperplanes are decision boundaries that help classify the data points. Data points falling on other side of the hyperplane can be attributed to different classes. SVM tries to find the "best margin" that separates the classes and reduce the risk of error on the data.



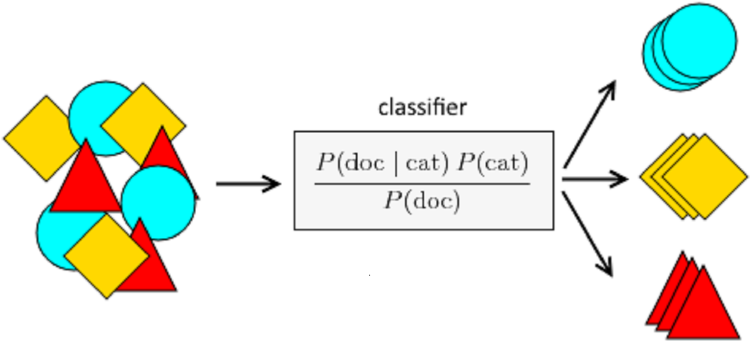
3. Multi-Layer Perceptron(MLP): A **multilayer perceptron (MLP)** is a class of feedforward Artificial Neural Network(ANN) which generates a set of outputs from a set of inputs. MLP uses Backpropagation for training which is a supervised learning technique. It has at least three layers of nodes: input layer, hidden layer, and output layer. It's input nodes are connected as a directed graph between input and output layers.



4. Random Forest(RF): Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. It builds multiple decision trees and merges them together to get a more accurate and stable prediction.



5. Naive Bayes(NB): Naive Bayes algorithm is a classification technique based on **Bayes’s Theorem.** It predicts membership probabilities for each class. The class with highest probability is considered as the most likely class.

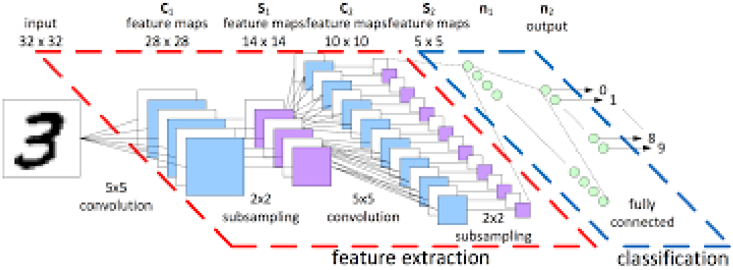


$$\frac{P(\mathrm{doc} \mid \mathrm{cat})\, P(\mathrm{cat})}{P(\mathrm{doc})}$$

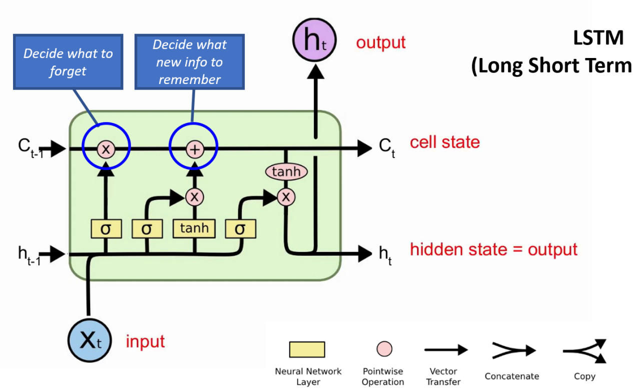
Model Implementation

To implement this model we used following Long Short Term Memory and Neural Network mechanisms.

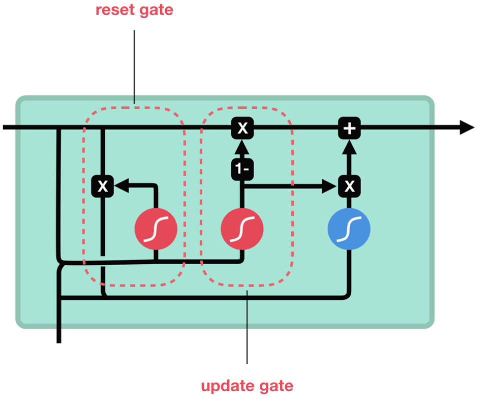
1. Convolutional Neural Network(CNN): Convolutional Neural Network is a class of deep neural networks used for NLP tasks such as Sentence and Text Classification, Sentiment Analysis etc.. A convolutional neural network is composed of "convolutional" and "downsampling" or "subsampling" layers.



2. Long Short Term Memory(RNN-LSTM): An LSTM is a special kind of Recurrent Neural Network(RNN) consisting of four neural network layers. These are recurring layers from the RNN with thre networks acting as gates: Forget Gate, Input Gate, Output Gate. An LSTM also has a cell state along with hidden state.



3. Gated Recurrent Unit(GRU): Similar to Long Short Term Memory(LSTM) Gated Recurrent Unit(GRUs) uses gates to control the flow of information. But GRU has only two gates, namely: Reset Gate(Short term memory), Update Gate(Long term memory0.These gates are two vectors which decide what information shouls be passed to the output.



4. Bidirectional RNN: Bidirectional RNN is just putting two independent RNNs together. It connects two hidden layers of opposite diections to the same output. This structure allows the networks to have both backward and forward information about the sequence at every time step. By this, the output layer can get information from past(backwards) and future(forwards) states simultaneously.

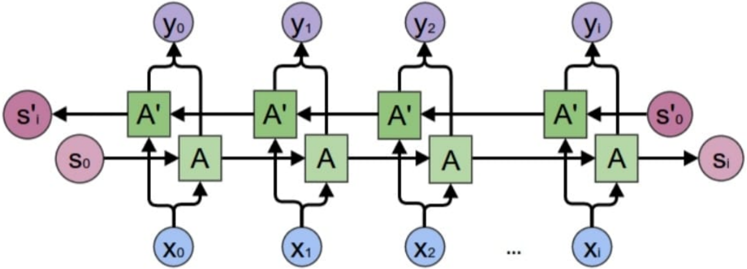
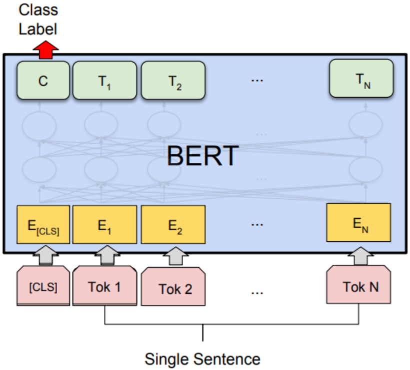


Figure: General structure of Bidirectional RNN

BERT:

Bidirectional Encoder Representations is a transformer-based machine learning technique. It is bidirectional because it is designed to read in both directions at the same time. BERT works similar to Transformer encoder stack, by taking sequence of words as input flowing up from one encoder to other.



Experiments and Results

1.Experiments: We performed experiments on given telugu dataset including Preprocessing, Training and Testing.

1.1 Preprocessing:

We removed all the punctuations and special characters present in the given question as input. Later, we divided the dataset into Training set and Testing(validation) set.

1.2 Model Training:

For training the model, the preprocessed data is given as input. Using training set we trained the classifiers such as LR, SVM, MLP etc..

1.3 Model Testing:

We used Testing set to test our model to find the accuracy of the model predictions.

2. Results: The **accuracies of the static classifiers** we used to train the model are shown in the below table:

Model	Word	Ngram	Char	Count vec
NB	73	73	77	76
LR	81	71	82	78
SVM	80	72	83	74
RF	80	76	84	77
DT	71	71	75	71
MLP	80	71	86	74

Deep Neural Network classifiers accuracies:

- CNN: 84
- LSTM: 83
- GRU: 84
- BiRNN: 84

Future Work and Conclusions

Future Work:

- We can use XLM-Roberta in BERT model which increases the accuracy of the model.
- XLM is a Transformer based architecture that is pre-trained using one of three language modelling objectives: Masked Language Modeling - the masked language modeling objective of BERT. Translation Language Modeling - translation language modeling objective for improving cross-lingual pre-training.

Conclusions: To train the model we implemented the following-

- Statistical classifiers such as LR, SVM, MLP, RF and NB.
- Deep Neural Networks such as CNN, RNN-LSTM, RNN-GRU, Bidirectional RNN and BERT.

References

Shivam Bansal
A Comprehensive Guide to Understand and Implement Text Classification in Python
April 23, 2018

Shraddha Anala
fastText for Text Classification
Nov 5,2020

Transfer Learning for NLP: Fine-Tuning BERT for Text Classification
Prateek Joshi, July 21, 2020