**CHAPTER 3**

**Methodology**

As discussed earlier, in the transliterated form of Bangla sentences, a lot of things are taken under consideration. In this form, concrete grammatical rules or hand-picked rules for the identification will not be enough for obtaining better accuracy and efficiency. The variation and informalities in the sentences are extremely high so therefore a learning-based approach is a must for addressing the problem in a greater extent. For this purpose, we have used supervised learning and deep learning. The rule based and learning based approaches are discussed below:

## **3.1 Rule Based Approaches**

Rule based approach is the conventional way to address the solution of the problem. With the combination of grammar and in-depth analysis of the dataset, we have designed a set of rules for the identification purpose. As the question mark is not that much significant and often gets omitted, we have excluded the presence of question mark as a rule. We have tried to design a set of rules for identification of Bangla interrogative sentences from the transliterated Bangla. The rules are as follows:

* *Rule 1*: A Bangla question word is present as the first or last word of a sentence.
* *Rule 2*: A Bangla question word is present as the second word of the sentence and the first word is the subject or object of the sentence.
* *Rule 3:* A Bangla question word is present just before the last word of the sentence.
* *Rule 4:* The word “*Naki*” is considered as a question word and the sentence with this word follows the previous rules.

As the designing and implementing the rule-based approach is not pragmatic, we have deviated ourselves from this approach and give our focus on the learning-based approaches.

## **3.2 Supervised Learning Based Approaches**

As mentioned earlier the identification can’t be done with accuracy and efficiency by only following a set of hand-picked rules. For better performance a learning-based approach is mandatory. We have implied the following supervised machine learning techniques for the classification of interrogative and non-interrogative sentences:

* Support Vector Machine (SVM)
* Logistic Regression(LR)
* Multilayer Perceptron(MLP)
* k-Nearest Neighbors (k-NN)

These machine learning algorithms are very common for text classification problems in Natural Language Processing(NLP). Our main challenge was to imply them in a dataset for transliterated Bangla sentences and extracting the ideal features.

**3.2.1 Support Vector Machine**

The Support Vector Machine invented by Vladimir Vapnik, is a very popular algorithm and is used extensively in a lot of sectors for supervised classification problems. In SVM, a hyperplane is drawn to separate the data points into two classes. The distance between the hyperplane and the points is called margin. If the margin is maximum from the closest data point, the hyperplane is called optimal hyperplane.

\*\*\* Figure of SVM

In our dataset we have two classes:

1. Interrogative Class
2. Non-Interrogative Class

We have found this very algorithm very suitable for classifying our dataset. Another noteworthy feature of SVM is the use of *Kernel* trick. With the help of this trick we can separate a linearly non-separable data points separable by mapping the data points in a very high dimensional space. There are four types of kernels which can be used in SVM. They are as follows:

* Linear Kernel
* Polynomial Kernel
* Sigmoid Kernel
* Radial Basis Function(rbf) Kernel

**3.2.2 Logistic Regression**

Logistic regression is a machine learning technique originally used in the field of statistics with a view to predicting the growth something in a limiting capacity environment. The output of this model can take only two values, 0 and 1.

**Logistic Function:**

Logistic function is also known as *Sigmoid* function. This is a S-curve representation which take any real number and map the number into the model in a range between 0 and 1. It is represented by the following equation:

f(n)= 1/(1+ e^-n)

n = the real number which is transformed

e = Euler’s Number

\*\*\*Figure of Logistic function

**Logistic Regression Equation:**

Logistic regression is somewhat analogous to the linear regression, but the main difference is that in logistic regression there are only two possible outcomes, 0 and 1. The equation for logistic regression is:

y = e^(b0 + b1\*x) / (1 + e^(b0 + b1\*x)) \*\*\*b= beta  
Here,

y = predicted outcome

x= input

b0= Intercept.

b1 = Coefficient for the input value x

Using the Maximum-Likelihood estimation algorithm the beta values are estimated from the training dataset based on some features.

**3.2.3 Multilayer Perceptron**

**Simple perceptron**

A perceptron or a single layer perceptron is the simplest form of a neural network where the weighted inputs are summed together by the activation function to check whether it can trigger the activation threshold value and generate the output. This output will be the input for another neuron. The neurons are independent to each other [12]. There are two layers in a simple perceptron, a) Input Layer b) Output Layer  
  
\*\*\*\*SINGLE Layer Perceptron Figure from ML book

**Limitation of Simple Perceptron**

But in 1969, Marvin Minsky and Seymour Papert demonstrate that the simple perceptron can only solve the linearly separable problems. A non-linearly separable problem like X-OR can’t be solved by a simple perceptron. To solve such problems, we need state-of-the art multilayer perceptron.

**Multilayer Perceptron**

In multilayer perceptron an additional layer is added named *Hidden Layer*. There can be more than one hidden layers in a multilayer perceptron. The total number of hidden layers in a multilayer perceptron depends on the nature of the problem. There are two challenges of using multilayer perceptron. One is for output layer the input is unknown and another is for output layer the threshold value of the activation function is unknown.

\*\*\*equation

**Back Propagation of Error**

* In each step calculate the error
* Propagate backwards
* Update the weights

**Error Function**

Error function is used to quantify the error. Most used function is the sum of squares function.

\*\*\*Error function: It’s in my Pad

**Training in MLP**

The training is done in two steps:

* Forward Step
* Backward Step

**Forward Step**

* The output of each hidden node is calculated using the input and the hidden node’s activation function.
* The output of the network layer/output layer is calculated by taking the hidden layer’s output as the input and the network layer’s activation function.

**Backward Step**

* The error of the network is found out using the error function.
* The hidden-to-output layer weights are updated using the error.
* Finally, the input-to-hidden layer weights are updated.

**3.2.4 k-Nearest Neighbors**

k-Nearest Neighbors is simple algorithm used for classification. The test data point is checked to find its k-nearest neighbors to determine the class of that data point. For k-NN the labeling of the training data points are not required.

\*\*\* Figure of KNN

This algorithm finds the distances of the k number of neighboring points. The following distances are used:

* Euclidean Distance
* Hamming Distance
* Minkowski Distance
* Manhattan Distance

Among them the mostly used is the *Euclidean* Distance.

**Limitation**

The efficiency and accuracy of k-NN attenuates with the enlargement of the dataset. With the increase of the dimensions, the k-NN classification faces the “Curse of Dimensionality”. With small dataset, k-NN works satisfactory.

**3.3 Deep Learning**

For classification and identification, using deep learning is a modern and dynamic approach. Deep learning provides paramount features and abilities for solving a great number of machine learning and artificial intelligence problems. For implying deep learning we have used Convolutional Neural Network (CNN) for classification.

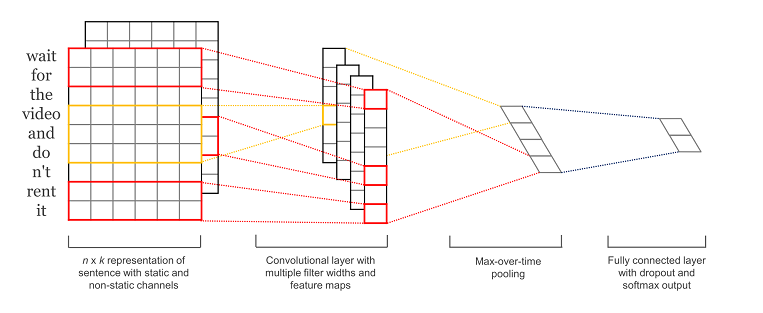
**3.3.1 Convolutional Neural Network**

CNN is a very popular artificial neural network model. CNN is extensively used in image processing, face recognition, video games, self-driving cars and many other sectors. In, CNN a number of hidden layers named as “Convolutional Layer” is present. To every convolutional layer there is one or more filters. The filters do the math behind the match of the patterns. Filter is a small matrix whose rows and columns are specified and the values within the matrix is initialized with random numbers. The deeper the CNN is the complex the filters get. We specify the number of filters a convolutional layer can possess. In CNN, every convolutional layer compute and detect features and the final layer gives the classification based on the detected features by the previous layers.

The CNN is also common in NLP. With the implementation of CNN, good results have been achieved in sentiment analysis, sentence classification and in spam detection. For implementing CNN, each sentence is represented as a matrix. Each word of the sentence is represented by each row of the matrix.

**3.3.3 CNN for Transliterated Sentence Classification**

For classification, we have tried to replicate the model of CNN described in [13]. This model is considered as one of the standard models for classification. The CNN of our model looks like the following figure:



We need to embed the words for CNN. The first layer is used to embed the words into low-dimensional vectors. This layer is a learning step. The embedded words act as a lookup table. The second layer is used for convolution. Embedded words are used for performing convolutions. The size of the filter varies with the length of the word and can take on multiple words at a time. The result of this convolution layer is max-pooled to get the feature vector and dropout regularization.