Chapter 4

Incorporating & Comparing Question Classification Accuracy for Generative Models

4.1 Introduction

The present chapter deals with the details about our research on a recognized dataset that has taken from TREC. The main aim of this research work is to find out the role of various features for the categorization of questions. For performing the experiment the questions which have been taken from TREC are 5500 questions used for training purpose and for testing 500 questions have taken.

The algorithm or the classifier used for classification purpose during this research work is Naïve Bayes and Back-Propagation Feed-Forward Artificial Neural Network (BPFFANN). Firstly the training phase is performed with 5500 datasets from TREC and afterwards the testing phase with 500 datasets.

4.2 A Generative Model for Question Classification

The research work started with very simple model Naïve Bayes and after then Back Propagation Feed Forward Artificial neural Network is used for the first phase for question classification task. For the combination of different feature set the experiment has carried out for both the classifiers and the comparative analysis has been presented in the current chapter.

4.2.1 Naïve Bayes Question Classification

A simple classifier which is based on the probabilistic model is Naive Bayes which is based on Bayes' theorem with the strong independence assumption. Bayes theorem can be stated as follows:

\[ P(\text{cl}|\text{l}) = \frac{\prod \left( \frac{\text{l}}{\text{cl}} \right)}{\prod} \]

(4.1)

Where \( P(\text{cl}|\text{l}) \) is the posterior probability.
P (cl) is the prior probability,
P (I|cl) is the likelihood
And P (I) is the evidence.

A Naive Bayes classifier go behind the provisional sovereignty because it suppose that the existence (or nonexistence) of a particular attribute of a class is distinct to the existence (or nonexistence) of any other attribute, specified for the class variable. Therefore, conditions are specified for a weight value which is autonomous for its location and occurrence of other conditions. Naive Bayes classifier is known as supervised learning classifiers because the training is given to the classifier by using some known examples.

4.2.1.1 Training a Naïve Bayes

The experiments for the classification of question category which is performed by Zhang & Lee was based on the Naive Bayesian classifier and was trained on the usual data set for question classification of Li & Roth, described in previous chapter[42].

Basically the model which was used by Zang and Lee was based on the bag-of-words features of a question that means it is used to characterize a question like an unstructured gathering of words. Inside this representation, a glossary is formed, whose length is identical to the amount of different terminologies available in the training set. Subsequent to that, questions can be programmed as binary feature vectors q, in such a manner that if the question includes the j-th word of the glossary, then qj is set to 1; otherwise, qj is set to 0. For demonstrating this representation we can take an example "What is the capital of India?" and "Where was Lord Krishna born?" and for this the resulting word glossary:

[What, was, is, where, the, Lord, Krishna, born, capital, of, India]

So the binary feature vector demonstration designed for the question "What is the capital of India?" would then be:

[1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1]
A different type of feature — bag-of-ngrams — was also employed in the research, in order to attempt the capturing of various dependencies among words, like the arrangement in which they come into view. By using bag-of-ngrams features, the classifier represents a minor enhancement in terms of exactness, over the bag-of-words features.

4.2.1.2 Naïve Bayes Training Algorithm

The algorithms proposed for carrying out this research work are preceded by exploring the feature vectors which are related to each class. Feature vector comprises general terms, which are going on questions, related to one exact class articulated in conditions of their weight or their significance in particular question. General terms are originated by the use of three questions related features that are: Lexical, Semantic and Syntactic. The algorithm started by stop-word elimination, stemming and then pruning (removing the words having the occurrences under a particular range and occurrences over a particular range). For calculating the weight of a word for any particular class, related to a particular question we can use weight calculation methods. For this work weight is computed by entropy, which is related with the word [59].

After that, the next task is to categorizing a new question. For this task the probability of the question belonging to all 6 coarse grain category and 50 fine grain category is calculated. The category for which we obtained the maximum posterior probability is the one for which the question is assigned means we can say that the question is belonging to that particular category.

\[ \text{cl MAP (I)} = \arg \max_{\text{cl} \in \text{CL}} P(\text{cl}/I) \]  
(4.2)

Where cl is the category with maximum posterior probability. The probability of a question I to belong to a category cl is given by eq. (5.1).

The accuracy was found to increase with increase in the number of training data. Matlab software has been used for the implementation Naïve Bayes algorithm.
4.2.1.3 Experimental Result & Discussion

Experimental Result

The experiment has been performed for the standard TREC 10 dataset in increasing order of questions which is shown in table 4.1. Started with 1000 questions, we can see from the table as we increase the number of question the question classification accuracy also increases. Naive Bayesian classifiers trained on 5,500 questions, perform its execution comparatively good for question classification task, with an accuracy of 80.6% for the coarse grained category and 78.6% for fine grain. Though, the outcome furthermore demonstrates that in order to accomplish this accuracy, the training sets require to be relatively large, because when the training is provided to the classifier for just 1000 questions the accuracy simply falls to 71.4% for coarse grain and 70.2% for fine grain.

Table 4.1: Classification Accuracy using Naïve Bayes algorithm for standard TREC 10 dataset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coarse Grain</td>
<td>71.4%</td>
<td>74.4%</td>
<td>75.0%</td>
<td>77.6%</td>
<td>80.6%</td>
</tr>
<tr>
<td>Fine Grain</td>
<td>70.2%</td>
<td>71.3%</td>
<td>72.6%</td>
<td>74.4%</td>
<td>78.6%</td>
</tr>
</tbody>
</table>

Discussion

Some additional analysis of the achieved results and discussion about how they can be improved is presented here. The main important characteristics of Naïve Bayes are:

1) They Work well on numeric and textual data.
2) Easy to implement.
3) Fast computation comparison with other algorithms.

But on the other hand the problem or the difficulty associated with the Naïve Bayes is:

1) If the conditional independence assumption is violated by real-world data, it performs very poorly
2) When features are highly correlated again, it performs poorly.
So despite the unrealistic independence assumption the Naive Bayes classifier is greatly effective in practice since its classification decision may often be correct even if its probability estimates are inaccurate. Because independent variables are implicit, merely the variances of the variables designed for each class required being determined and not the whole covariance matrix. So for the further improvement of the question classification accuracy, we take Artificial Neural Network, a potential reason to pick artificial neural networks (ANN) over Naive Bayes are:

1) Correlations between input variables.
2) Naive Bayes assumes that all input variables are independent. If that hypothesis is not accurate, then it can influence the correctness of the Naive Bayes classifier and the maximum numbers of errors come from the OTHER fine-grained category, of each coarse-grained category.
3) An ANN with appropriate network structure can handle the correlation/dependence between input variables.

4.2.2 Question Classification using BP-FFANN Classifier

It is very difficult to classify the question and to find out the actual category of question for question answering system when there is a huge amount of questions. In this research work for classifying the questions supervised machine learning classifier has used. But the problem associated with this task is that to find out or to accommodate the rich set of features for different category of questions and this process is known as feature engineering. During this effort the main concentration of this research work is to explore the three main features of a question that are the lexical, semantic and syntactic. By the help of this training of the classifier has performed and that can make affluent information support. The thesis proposes here the BPFFANN (Back Propagation Feed-Forward) algorithm for classifying the question and that can be efficiently used by the question classifier [60]. Li and Roth’s two-layer question type taxonomy has been used for carrying out this research, consisting of 6 coarse grained categories and 50 fine grain categories and has already described in previous chapter [51]. The questions for training and testing the Back Propagation Feed-Forward classifier has taken from TREC (Text Retrieval Conference) which is created by Li and Roth and publically available for testing and
training purpose [51]. The assumption considered here is that each question will be belonging to its own group or category. The overall conversation comprises in the following sections of this chapter of the thesis and the work has been done with feature extraction technique using artificial neural network feed forward back propagation hypothesis.

4.2.2.1 Structure of Knowledge-Based Model of Questions

The formation of a knowledge-based model of different domain related question consists of various steps that have been depicted in figure 4.1.

![Figure 4.1 Framing of Knowledge Based Model](image)

**4.2.2.2 Representation of Neurons**

Neuron representations are very significant in Artificial Neural Network. The basic constituents of Neuron are Input, Weights, Adder and Activation Functions:

**Inputs, Weights and Adder Function:** Artificial Neural Network is a combination of inputs there weight and the adder which provide the summation of input with their respective weights. Let the neuron inputs be (I1) and there weights are Wt1, Wt2... Wtn and the adder function which is the sum of the weights of the different inputs are:
Adder (Sum) \text{ weighted} = \sum W_t_i * I_i \\
(4.3)

Where \( n \) = the number of inputs

The output neuron is calculated by the activation function which is useful for preventing the amplitude of the neuron. Let the output of neuron is \( (O) \) and the activation function is \( Af \) then the output neuron is calculated by the formula:

\[
O = Af(\text{Adder (sum) weighted} + \tilde{U}) = Af(\beta)
\]
(4.4)

Where \( \beta = \text{Adder (sum) weighted} + \tilde{U} \)

Here \( \tilde{U} \) = denotes bias

\( \phi \) = the induced field of neurons.

The bias \( \tilde{U} \) has the consequence of concerning a transformation to the weighted sum and it is an exterior restriction of the neuron. The above expression is redrafted in the following manner.

\[
\beta = \sum W_i * A_i , \text{where } W_{t_0} = \tilde{U} \text{and } I_0 = 1
\]
(4.5)

It can be represented in diagrammatic form as shown below in Figure 4.2 it shows the structural design of neurons with input \((I_1, I_2, \ldots, I_n)\) and output \(O\) acquired subsequent to concerning the activation function on the weighted sum as given the details above.
Optimal Transfer Function: It is also known as activation function, it is the function that resolves the structural design of a neuron model. For this work we employ the TANSIG tan-sigmoid function as an optimal transfer function.

4.2.2.3 Structural Design of Feed-Forward-Artificial Neural Network (FFANN)

Multi layer feed forward artificial network construction has been used during this research work. Basically the use of Multi layer feed forward artificial neural network (FFANN) is in that situation where it is difficult to split the data linearly even if also by a hyper plane. For this research work, two layers of FFANN has used, actually it contains hidden layers between input and output layers which is not responsible directly for getting input from input domain or sending output to the outside surroundings. For this research work we make use of 2 † layer FFANN including 10 hidden neurons.
4.2.2.4 Artificial Neural Network Learning Method

Generally the technique used for classifying any data set is back-propagation feed forward. In this research work the method used is: Levenberg-Marquardt optimization method as a training function. The reason behind choosing, this is that it is the most suggested methods in supervised learning algorithm. The network training function that is TRAINLM is used in this work that revises the weight and bias values according to the Levenberg-Marquardt optimization. In Matlab the fastest back propagation training algorithm which is used by many researchers are based on the Levenberg-Marquardt optimization method and it is in nn-tool box in Matlab.

4.2.2.5 Back-Propagation Feed- Forward Artificial Neural Network Algorithm

For nonlinear regression function or for classification of patterns in supervised learning which is using the technique of Multi layer feed forward network can be trained using a set of input example of the network behavior that is network input and target output.

Back propagation Feed-forward algorithm consists of the frequent functioning of following two phases. Feed-Forward phase: for this phase, input is provided to the network and error is calculated for each neuron at the output layer. Back-Propagation phase: in this phase the network error is calculated and is used in revising the weights and broadcasted from the last node to start node represented in figure 4.3.

![Flow Diagram of Back Propagation Algorithm](image)

**Figure 4.3:** Flow Diagram of Back Propagation Algorithm
Performance Function: During the procedure of training artificial neural network it engages changing the assessment of weight and training function (activation function) for optimizing the performance of the network. By default the performance function for feed forward network is mean square error (MSE). \( (e_i)^2 \) is the error between the output of a network \( O \) and target output \( TAR \)

\[
(e_i)^2 = (TAR_i - O_i)
\]

(4.6)

\[
MSE = -\sum (e_i)^2
\]

(4.7)

\[
MSE = -(TAR_i - O_i)
\]

(4.8)

Adaption Function: Adaption function is mainly used for updating the weight of the network and biases.

LEARNGDM - LEARNGDM (Gradient Descent with Momentum weight and bias learning function) used as adaption function for this research work. The use of LEARNGDM is for updating the weight of network and biases in the direction where the performance functions declining most rapidly, the -ve of the gradient. For e.g. one of the iteration

\[
A_{i+1} = A_i - R_i G_i
\]

(4.9)

Where \( A_i \) = is the vector of current weight

\( G_i \) is the current gradient

\( R_i \) is the learning rate

The above equation is repeated until the network converges.

Training Function: Training functions generally used for training the network including there inputs, weight of neuron and transfer function. In this research work,
TRAINLM function is used for training algorithm that can train any network as well as its weight, net-input, and transfer functions that is actually derivative functions. Back propagation is used to calculate the Jacobian $jA$ of performance perf with respect to the bais and weight variables $A$. Each variable is adjusted according to Levenberg-Marquardt.

\[
\begin{align*}
jj &= jA \ast JA \\
je &= jA \ast E \\
\frac{dA}{je} &= -(jj + I_d^\ast\mu) \backslash je
\end{align*}
\]

Where $E$ =is all errors
$I_d$ =is the identity matrix.
The adaptive value $\mu$ is amplified by $\mu_{inc}$ until the above change results in a reduced performance value.

**Algorithm 4.1**: Training Algorithm for BPFFANN

**Step 1**: Initialize any variable $I=1$; $W (I)$ arbitrarily
**Step 2**: while (stopping criterion is not satisfied or $I< max\_iteration$)
**Step 3**: intended for every input $A$
**Step 4**: perform the Execution of the network with input $A$ and later on calculating the value of output $B$
**Step 5**: perform the updation to the weight of neurons in the back-word order using the below mentioned formula:
(Vector of the current weight)$_{i+1}$=(vector of the current weight)$_i$- (learning rate)$_i$(current gradient)$_i$.
**Step 6**: $I=N+1$
**Step 7**: end-while

4.2.2.6 Experiment

**Acquirement of Input Question**

Employment of the dataset for this research effort is the one which is produced by Li and Roth [51]. They make available a question dataset which is commonly utilized in question classification learning and known as trec (text retrieval conference)
dataset or publicly available dataset of the University of Illinois at Urbana-Champaign (TREC 10). Basically TREC data sets are the constitutions of 5500 tagged question which is used as training set and 500 autonomous tagged questions for the conduction of the test. Basically the data sets are the text files containing the label for each question in every row snapshot of TREC data set is presented in figure 4.4. The following is a snapshot of the TREC training set.

```
DESC:manner How did serfdom develop in and then leave Russia ?
ENTY:cremat What films featured the character Popeye Doyle ?
DESC:manner How can I find a list of celebrities ' real names ?
ENTY:animal What fowl grabs the spotlight after the Chinese Year of the Monkey ?
ABBR:exp What is the full form of .com ?
NUM:ind What contemptible scoundrel stole the cork from my lunch ?
NUM:gr What team did baseball 's St. Louis Browns become ?
NUM:title What is the oldest profession ?
DESC:def What are liver enzymes ?
NUM:ind Name the scar-faced bounty hunter of The Old West .
NUM:date When was Ozzy Osbourne born ?
DESC:reason Why do heavier objects travel downhill faster ?
NUM:ind Who was The Pride of the Yankees ?
NUM:ind Who killed Gandhi ?
ENTY:event What is considered the costliest disaster the insurance industry has ever faced ?
LOC:state What sprawling U.S. state boasts the most airports ?
DESC:desc What did the only repealed amendment to the U.S. Constitution deal with ?
NUM:count How many Jews were executed in concentration camps during WWII ?
DESC:def What is ` Nine Inch Nails ` ?
DESC:def What is an annotated bibliography ?
NUM:date What is the date of Boxing Day ?
ENTY:other What articles of clothing are tokens in Monopoly ?
NUM:ind Name 11 famous martyrs .
DESC:desc What is the Olympic motto ?
DESC:desc What is the origin of the name ` Scarlett ' ?
ENTY:letter What 's the second-most-used vowel in English ?
NUM:ind Who was the inventor of silly putty ?
LOC:other What is the highest waterfall in the United States ?
ENTY:other Name a golf course in Myrtle Beach .
```

**Figure 4.4** Snapshot of TREC Question Data Set

For this research work two categories of questions that are coarse grain and fine grain categories have been used, which have already explained in earlier chapters. Coarse and fine grains are two layer architecture and it is the combination of 6 coarse grained classes and 50 fine grained classes. Figure 4.5 shows the front view of the classification process and figure 4.6 shows input question sets.
4.2.2.7 Evaluation Measures

For the evaluation of system performance there are so many evaluation measures that have been already proposed in various literatures. In this research work accuracy is used for evaluating the performance of the system. Right through the years, accuracy has been applied as the most important assessment for the evaluation of systems contributing in the QA way [61], [62], [63]. Accuracy can be defined as the division of the total no of correctly classified question to the total no of input questions.

\[
\text{Accuracy} = \frac{\text{No. of Question Correctly Classified}}{\text{Total Number of Input Question}}
\]
4.3 Feature Extraction

The present segment deals with the study and feature mining effort of the proposed work. The basic three categories of features regarding any kind of question are extracted here subsequent to amalgamation of all the features the information database is constructed for the question. The steps are as follows:

4.3.1 Lexical Feature Extraction of Questions

In the context of lexical feature extraction, the primary concern is about the stopword and stemming deduction procedure and approximately 2000-2700 stopword are deduced. After then the stemming that is the removal of the grammatical unusual derivation of the verb is performed. Later than after removal of stopword and stemming the output hold a smaller amount of dimensional elevated characteristic
information which is at last combined with additional features. The following figure 4.7 shows stopword/stemmer removed output.

![Stop and Stem Window](image)

**Figure 4.7:** Output after Lexical Feature Extraction (stop-word/stemmer removal)

### 4.3.2 Syntactic Feature Extraction of Questions

In this research work three subcategories of syntactic features are extracted which is already covered by us in the earlier chapters. For designing the development of syntactic feature extraction the diverse open cause standard is used to carry out transitional press.To take out all syntactic related feature of the question it is the primary prerequisite to generate the parse tree for the input question subsequent to parsing all the parsing consequences are accumulated. For designing the development of parsers Stanford Lexicalized Parser v3.2.0. has used. One of the parse trees of the input question is shown in figure 4.8.
4.3.2.1 Headword Extraction of Questions

For extracting the headword algorithm 3.1 has used, the output generated by the Stanford parser and it takes out the head word of the question. Headword plays the significant role in the question classification because it is used to allocate the class of question. The figure 4.9 shows the extracted headword of the question, which is used to assign the category of question.

4.3.2.2 Question Prototype Extraction of Questions

For finding out the question prototype algorithm 3.2 has used, which is helpful for judgment of the question class of the input question to formulate a prosperous information support. The input question given is classified into the special question categories according to Li & Roth's two-layer organized classification. The figure 5.10 shows the output of question pattern, prototype; here the output 'other' means the question not matched with any category.
Figure 4.9: Extracted Headword of the Question

Figure 4.10: Output Extracted Question Category on the Basis of Question Prototype
4.3.2.3 Part of Speech (POS) Extraction of Questions

For the potential utilization of the judgment of the class information support, component of the words has taken that is the part of speech. Which in fact encloses a variety of elements of parser created passage ingredients. This is making the use of discovering the syntactic classes of the input questions. The figure 4.11 shows the output of POS extraction process.

4.3.3 Semantic Feature

At last semantic feature extraction has performed. How the extraction of semantic feature has been performed discussion about it is given in subsequent section and also discussion about the various sub features and their extraction is mentioned.

![Figure 4.11: Output of Parts Of Speech Extractor](image)
4.3.3.1 Name Entity Recognizer Extraction of Questions

For supporting the mining of Name Entity, Stanford Name Entity recognizer (NER) V 3.2.0. has used. Which acquire every question and categorize it into 7 diverse classes. Figure 4.12 shows the output of Stanford NER MUC7 for the input questions and figure 4.13 shows the Output of Name Entity Extractor. Seven diverse classes are as follows:

1. LOCATION
2. TIME
3. PERSON
4. ORGANIZATION
5. MONEY
6. DATE
7. PERSON

![Stanford Named Entity Recognizer](image)

**Figure 4.12:** Output of Stanford NER MUC7 for the Input Question
4.3.3.2 Semantic Headword Extraction of Questions

For extraction of semantic significance, it is extremely essential to take out the headword which includes all semantic meaning of the question. For mining of diverse synonyms, hyponyms and hypernyms of the statement the WordNet 2.1 and algorithm 3.3 has used, which is in fact the record includes all information. By means of applying the WordNet it can take out the semantic headword of the question and on the stand of diverse statement, it produces the dissimilar fine grain group for classifying the question. The figure 4.14 shows the output of the semantic headword extractor and null value shows that the question is not belong to any WordNet selected category.

4.4 Creation of Information Support Replica

Subsequent to the improvement of passage characteristic the research work proposes the information Support Replica. The investigational arrangement of structuring the information-base is performed how it is performed is already conversed in the above successive section of this chapter. For supporting the generation of Information support replica the proposed work just combine all three feature discussed and get the output on the basis of it for different fine grain and coarse grain category of the question represented in figure 4.15. And the output of information support replica is shown in the figure 4.16

![Output of Name Entity Extractor](image.png)
Figure 4.14: Output of Semantic Headword Extractor

Figure 4.15 The Structure of Information Support Replica
4.5 Gathering and Training of the Data for BPFFANN Classifier

The preparation or the arrangement is created which includes the diverse fine grain associated grouping of the input question. For carrying out this research work the unigram (bag-of-word) representation of feature has employed for categorizing the questions. Integer information for every fine grain category has been used and here the classification of question is performed on the basis of 50 fine grain categories and 6 coarse grain categories.

![Information Support Replica after merging all Three Features](image)

4.5.1 Construction of Network for ANN

**Input data setup** - For the input setup two kinds of data stream have been used that are: training data and testing data. Training data are used for providing training and testing data are used for testing purpose.

Output data setup- For accumulating the output data every time Tar is used as target data.
Network for ANN- Feed-Forward-Back-Propagation network category has been used for performing our research work. In this arrangement two data streams have used for training purpose that is the input data and the target data. Some essential elements for construction of network mentioned below:

i. TRAINLM function is used which is previously conversed.

ii. MSE-for the calculation of error performance.

iii. TANSIG sigmoid transfer function is used

2 layers FFANN with the 10 number of neurons has used.

Input data setup- Two input data Train and Test has been taken for training and testing of the network. Output data setup- For storing the output data each time Tar has used as target data. As shown in figure 4.17.

4.5.2 Training Phase of the Network

In the first phase of providing the training 1000 question has been taken for the training and testing. Training is provided to the classifier for the first 900 question with the utilization of the extracted combination of 3 features for both coarse and fine grain. And subsequent to training, network is replicated with 100 (901 to 1000) questions to find out the course grain and fine grain category of the input test questions for justification and to find out the accuracy of the algorithm throughout the training. The development in the training set is continually restructured for the training window. The extent of inclinement of presentation and the whole number of confirmation is checked. These (The extent of inclinement of presentation and the whole number of confirmations) are used to stop the training process. If the slope turns out to be small, training arrives at a small amount of the performance, if the total magnitude of gradient is less than 1e-5, the training will discontinue.

Firstly the training and testing is performed for 1000 questions subsequently it is increased to 2000, 3000, 4000 and for 5500 questions. And at last training is provided for 5500 questions and testing is done by 500 separate questions provided by TREC.
Figure 4.17: Neural Network Active Window

4.5.3 Post Training Scrutiny of the Network

The following figure 4.18 indicates that the iteration at which the validation performance reached a maximum was 78. The validation and test curve are almost same.

The figure 4.19 demonstrates the regression graph of the data. (The three axis correspond to the training, validation and testing of the data the dashed line represent the perfect result (output → target) the solid line is used to represent best fit linear regression between output and targets. And the R =1 represent the exact relationship between output and target).
Figure 4.18: The Performance Validation Graph

4.5.4 Experimental Result

After training phase last 100 questions has used for testing the network and store the data in TAR target data using BPFFANN classifier.

Figure 4.20 demonstrate that after combining the various lexical and syntactic feature for 1000 questions the accuracy result is greatest when combine the unigram, stemming stop ward, headword, question prototype and part of speech features. Accuracy reaches to question classification is 84.2% for coarse grain classification when combine the unigram, stemming stop ward, headword, question prototype and part of speech features. Similarly for fine grain classification accuracy is represented in figure 4.21. It is observed that for the combination of unigram, stemming, headword and part of speech result is better as compared to other combinations and it gives the 83% accuracy for fine grain.
Figure 4.19 the Regression Graph of Training

Figure 4.22 represents that the combination of three features, lexical, semantic and syntactic that are: the unigram, stemming, headword, question prototype, named entity recognizer and semantic headwords provides the accuracy 86% for coarse grain for 1000 questions. In figure 4.23, accuracy reaches 84.8% for fine grain when unigram, headword, question prototype, named entity recognizer and semantic headword features combined that are extracted from lexical, semantic and syntactic features.

All the experiment performed above is repeated for increasing no. of questions which is shown in figure 4.24 and figure 4.25 for coarse and fine grain respectively, and it can observe that the highest accuracy for coarse grain is 93.2% and for fine grain is 89.2% for the setup of 5500 training questions and 500 test questions. Comparative observation has shown in figure 4.26 for coarse and fine grain classification.
**Figure 4.20** Combination of Lexical and Syntactic Feature and their Accuracy for Coarse Grain for 1000 Questions (using BPFFANN Classifier)

- 1 - Unigram
- 2 - Stemming
- 3 - Stopword
- 4 - Headword
- 5 - Question Prototype
- 6 - Part of Speech

**COARSE GRAIN**
*for 1000 Questions (Lexical,Syntactic)*

<table>
<thead>
<tr>
<th>Features</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>80%</td>
</tr>
<tr>
<td>2,3,4</td>
<td>83%</td>
</tr>
<tr>
<td>1,3,4,5,6</td>
<td>84.2%</td>
</tr>
</tbody>
</table>
Figure 4.21 Combination of Lexical and Syntactic Feature and their Accuracy for Fine Grain for 1000 Questions (using BPFFANN Classifier)
Figure 4.22 Combination of Lexical, Syntactic and Semantic Features and their Accuracy for Coarse Grain for 1000 Questions (using BPFFANN Classifier)
Figure 4.23 Combination of Lexical and Syntactic Feature and their Accuracy for Fine Grain for 1000 Questions (using BPFFANN Classifier)
Figure 4.24 Accuracy of 5500 Questions for the Combinations of Lexical, Semantic and Syntactic Feature for Coarse Grain (using BPFFANN Classifier)

Figure 4.25 Accuracy of 5500 Questions for the Combinations of Lexical, Semantic and Syntactic Feature for Fine Grain (using BPFFANN Classifier)
4.5.5 Results Analysis & Discussion

Result Analysis

After the completion of training phase the result obtained as classification accuracy by performing the testing phases. The accuracy is calculated by the below mentioned formula:

\[
\begin{align*}
\text{Accuracy} &= \frac{\text{Correct Predictions}}{\text{Total Predictions}} \\
&= \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}}
\end{align*}
\]

For 1000 questions training and testing is provided and for the different combination of features which have mentioned earlier, the result obtained as accuracy 86% for coarse grain classification 84.8% for fine grain classification. The comparative result is presented in fig 4.6 for 5500 training questions, coarse grain classification

---

Figure 4.26 Comparison of Accuracy for 5500 Questions for the Combinations of Lexical, Semantic and Syntactic Feature of Coarse Grain and Fine Grain (using BPFFANN Classifier)
accuracy is 93.2% and fine grain classification accuracy is 89.2%.

**Discussion**

The result obtained by BPFFANN for the question classification accuracy is 93.2% for coarse grain, which is outstanding than the 80.6% which is obtained from the Naïve Bayes classifier. Similarly for fine grain the classification accuracy for BPFFANN is 89.2%, which is again much higher than the Naïve Bayes, which gives 78.6% accuracy. The result generated by BPFFANN is quite good and having the following advantages for question classification:

1) In general, forecasting precision is usually high as compared to Bayesian methods.

2) Strong, working when the training model includes errors.

3) Quick estimation of the learned target functions, whereas Bayesian networks is usually time-consuming.

But on the other-hand the problem that has been encountered during the classification work which is related with BPFFANN are:

1) It takes extensive time for training.

2) It is very hard to know about the learned function (weights).

3) Tough to integrate domain knowledge.

4) ANNs frequently converge on *local minima* rather than global minima.

5) ANNs repeatedly suffers the problem of *over-fitting* if training goes away excessively long, the significance is that for any particular pattern, an ANN might initiate to believe the noise as part of the pattern.

So the alternative is the support vector machine (SVM) because SVMs don't suffer from either of these two problems that are: local minima and over-fitting. And SVM is also useful on large database. Because the complication of trained classifier is differentiated by the no. of support vectors rather than the dimensionality of the data. The support vectors are the necessary or significant training examples they lie closest to the decision boundary. If every additional training example is detached and the
training is continual, the identical separating hyper plane would be originated. The quantity of support vectors originated can be utilized to calculate an (upper) bound on the anticipated error rate of the SVM classifier, and it is not dependent on the dimensionality of the data. Therefore, an SVM with a minimum amount of support vectors can have good quality generalization, even while the dimensionality which is used for the data is high.

4.5.6 Summary

The present section deals with result obtained by: Naïve Bayes classifier and Back Propagation Feed Forward Artificial Neural Network (BPFFANN). In this section Information support replica has been created by making an allowance for the significant feature of input questions. After the gathering of information design, construction of networking has been performed. After construction of the network training phase, subsequent to that testing phase is also explained. The study and outcome have been observed and compared for both the classifiers NB and BPFFANN. It is observed that the implementation and computation of NB is easy as compared to BPFFANN whereas in BPFFANN forecasting precision is high, working of training model is strong even if training model includes errors for question classification. NB is time consuming than BPFFANN for question classification task.

4.6 Enhancing the Generative Model

Question classification is extremely essential for question answering system. For removing the drawbacks of BPFFANN which have been already discussed in the previous chapter and for improving the question classification accuracy which means for improving the generative model, the experiment has been performed with Support Vector Machine (SVM) using the different combinations of question feature-set.

As similar to the previous chapter for performing the research work the dataset is taken from the TREC QA track. Firstly the training phase is performed with 5500 datasets from TREC and afterwards the testing phase with 500 datasets.

4.6.1 Support Vector Machine
The Support Vector Machine (SVM) is a promising machine learning tool and it is also very valuable alternative for the technologies which is based on naïve bayes, the nearest neighbor or neural network [64]. A well known supervised learning algorithm for doing research, especially in the field which is related to the discovering of numerical practices, Support Vector Machine was established in 1970s and was established by Vladimir Vapnik and was promoted in the nighties by Vapnik and colleague [65]. Essentially the SVM is a learning technique which is based on the supervised machine learning algorithm and the training is provided to the SVM by using a dataset that is used for the classification purpose and includes attribute sets with both positive and negative instances. The SVM performance or its functioning is basically based on the mapping of the training dataset into a feature space is represented in Figure 4.27 and it computed the most favorable hyper plane, which is also known as decision boundary which will divide the positive and negative characteristic points in such a way that will maximize the space or margin from every classification margin or support vector. SVM is based on the concept of maximization of the margin that is the difference between the two classes, simultaneously minimizing the misclassification of data, axis X1 and X2 represents the feature set for the classification [66].

There can be, however, more than one such hyperplane, if such type of condition occurs, there is choice arises for SVM to choose one for which the distance between it and the nearest training instances in either side is maximized. This distance can be described as the margin, and such hyper plane is usually referred to as the hyper plane having the maximum margin. Naturally, it is anticipated that the better the margin, the better the model will generalize to unseen examples.
The difficulty related to the open-ended question categorization is directly associated with the text classification problem, inside that machine learner have to classify documents that may or may not present a general overlap of keywords or key expression that is used to set up a document group association. The utilization factor of an SVM in text classification has been healthy researched and has established for high-quality presentation, excluding the need of compound NLP [67]. The work out of an SVM for a text classification difficulty has the need of significant weights to be allocated to every word in the corpus of exercise documents - resulting an extremely dimensional feature set.

Some research exhibits term-to-vector conversions and evaluates the presentation of word-stemming against the non-stemming features using different weighting aspects like: inverse document frequency and redundancy which is used as input to an SVM for document retrieval [68]. They demonstrate that if the word stemming feature is used along with Term-Frequency-Inverse Document Frequency (TF-IDF) it will reach to its superior performance in document recall through a analogous SVM.

It may be possibly disputed that an easy keyword exploration approach can give the performance identical, or enhanced than that of machine learning techniques. Though, this would indicate a remarkable quantity of effort and it do not take the consideration of the fact that delicacies may survive in the problem domain that
formulate simple text parsing and keyword identification [42]. Furthermore, the SVM has been exposed to contain the greatest classification precision in additional question classification problems which is uneven against many other machine learning algorithms like Nearest Neighbor and Naïve Bayes [42].

Selection of the most appropriate kernel highly depends on the problem at hand and fine tuning its parameters can easily become a tedious and cumbersome task. The choice of a Kernel depends on the problem at hand because it depends on what we are trying to find out as a result. For example, if we are using Radial basis kernel it is helpful to select hyper spheres, whereas linear kernels use full for hyper planes. The inspiration for the selection of kernel is straightforward depends on which sort of information is to be expected to take out from the data.

For performing the classification by using SVM classifier, a freeware Support Vector Machine application termed as SVMLib has been utilized. SVMLib application has four offered kernels which are built in are: linear, polynomial, radial basis function, and sigmoid. Similar to several other learning algorithms, Support Vector Machines need parameter tuning in order to get better precision of their test results.

4.6.2 Kernel Functions

The basic 4 types of kernel functions for SVM are: linear, polynomial, sigmoid and radial basis function among which linear is best for classification task [46]. Here is a list of some kernel functions accessible in SVMLib.

4.6.2.1 Linear Kernel

Different from the other kernels linear kernel require tuning only one parameter that is penalty parameter C. It is an unusual case of the Radial Basis Kernel apart from the other four kernels. This kernel achieves fine if the amount of features is large and it is not necessary to map to a upper dimensional space. Though it treats inadequately, with "noisy data"[46].

\[ K(x_i, x_j) = x_i^T x_j \]  

(4.13)

4.6.2.2 Polynomial Kernel
The polynomial kernel requires tuning of more parameters as compared to others which are available in SVMLib application. Adding together the fundamental parameters that are: penalty parameter C and Gaussian kernel function $\gamma$, furthermore, it has two additional parameters; the polynomial degree $d$ and the degree coefficient $r$. It is important to be reserved for the selection of $d$, as kernel values, possibly will go to infinity or zero when the value of $d$ is highly increased and polynomial kernel is used in machine learning algorithm for feature interaction [69].

$$K(x_i, x_j) = (\mathbf{x}_i^\top \mathbf{x}_j + r)^d, \quad C > 0$$

(4.14)

4.6.2.3 Radial Basis Function (RBF) Kernel

In a non-linear fashion, mapping of examples into the upper dimensional space is performed by RBF kernel. So just contrasting the linear kernel, RBK kernel can be able to manage such type of cases where the nonlinear relationship between the class labels and attributes. Though, unlike the Linear Kernel, it isn't appropriate while the number of features is "exceptionally large" [70].

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad C > 0$$

(4.15)

4.6.2.4 Hyperbolic Tangent (Sigmoid) Kernel

The Sigmoid Kernel, though an admired Support Vector Machine kernel usually used in neural networks, has numerous belongings that remain not completely studied. It is known, however, that unlike the other Support Vector Machines Kernels available in SVMLib, the Sigmoid Kernel is only conditionally a positive semi-definite (PSD) function for only some combinations of its free parameters (slope and bias of the function) [71]. Generally sigmoid kernel is used for neural network field. It is often used as neuron activation function. Non-PSD kernels can't be separated into an inner product form, they're not better than radial basis function kernels (RBF); in favor of
This reason, guides of using SVMLib provide an advice that this kernel is not suitable for a number of parameter choices [72].

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$$  \hspace{1cm} (4.16)

For sigmoid kernel two adjustable parameters that are: the slope $\gamma$ and the intercept constant $r$.

### 4.6.3 Training the Support Vector Machine

Again for providing the training to the support vector machine three features of a question set that are: lexical, semantic and syntactic features are added for classification of a question. The features and their extraction mechanism have already mentioned in previous chapter. Since this is often a supervised learning approach, so that's why the training is provided with the questions having pre-assigned categories. First of all, of the 1000 TREC questions training and testing is performed by using support vector machine method for the classification of question into fine grain and coarse grain categories. Subsequently, it is increased to 2000, 3000, 4000 and for 5500 questions, and at last training is provided for 5500 questions and testing is done by 500 separate questions provided by TREC.

Basically TREC organization provided 5500 questions set for training and 500 question set for testing, which have already stated in previous chapter. How the knowledge mining is performed for SVM is demonstrated in figure 4.28, similar as the previous chapter, in which knowledge mining is performed for neural network like the similar way, mining of the different question related features has performed for SVM. For figure 4.28 different representations are:

- S: Stemming Feature
- SPR: Stopword Removal
- QHE: Question Headword Extraction
- QPE: Question prototype Extraction
- POS: Part of Speech
- NER: Named Entity Recognition
SHE: Semantic Headword Extraction

**Figure 4.28** Proposed System Structure

### 4.6.4 Dataset

For carrying out this research work the publicly available training and testing datasets has been used which is provided by TREC 10 and TREC which have already mentioned in previous chapter. All these datasets have been manually labeled by TREC 10 according to the coarse and fine grained categories and represented in appendix A.

There are about 5, 500 labeled questions randomly divided into 5 training datasets of sizes: 1000, 2000, 3000, 4000 and 5500 respectively. The testing dataset contains 500 labeled questions from the TREC10 QA track.

### 4.6.5 Features

In this research work again three categories of features have been taken which is related to the question set, which have been mentioned in previous chapters that are:

i. Lexical Feature
ii. Semantic Feature

iii. Syntactic Feature

Lexical features: Lexical features are word related features. Three lexical features that have been taken for our research work that are: Bag-of-gram features, Stemming and Stopword Removal.

Syntactic features: Syntactic features are the features related to the grammatical structure of a question. The syntactic features that have taken for this work are:

i. Question Headword Extraction

ii. Question Prototype Extraction and

iii. Part of Speech (POS) Extraction

Semantic features: The features that are related to the meaning of the words in the question. Two semantic features have been used that are: Named Entities and Semantic Headword (word net).

Extraction of different word related features for training the SVM classifier for example the algorithms and the software that are used are same as the previous work which have been already mentioned in previous chapters.

4.6.6 Algorithm

Suppose a given training set \((x_i, y_i), i = 1, ..., n\), in which \(x_i = (x_{i1}, ..., x_{id})\) is a \(d\)-dimensional sample and \(y_i \in \{1, -1\}\) is the corresponding label. The task of a support vector classifier is to find a linear discriminant function \(g(x) = w^T x + w_0\), such that \(w^T x_i + w_0 \geq +1\) for \(y_i = +1\) and \(w^T x_i + w_0 \leq -1\) for \(y_i = -1\).

Hence the optimization problem will be:

\[
\text{Min} -w^T w - \sum \alpha_i \left( y_i (w^T x_i + w_0) - 1 \right) + C \sum \xi_i \quad (4.17)
\]

Such that
\[ y_i( w^T x_i + w_0 ) \geq 1 - \xi_i, \quad i = 1, \ldots, n \]  

(4.18) and

\[ W = \sum \alpha_i y_i x_i \]  

(4.19)

Where \( \xi \) is a positive slack variable and \( w \) is a weight vector.

Where \( \{ \alpha_i \mid i = 1, \ldots, n; \alpha_i \geq 0 \} \) are Lagrange multipliers or support vectors and we will be taking \( C \) as a penalty parameter for error calculation on the training set. For the linear separation of data, typically the feature space should be mapped to a higher dimensional space. The mapping is done by a so-called kernel function. The simplest type of linear kernel for two question \( x_i \), and \( x_j \) is defined as follows:

\[ K_{\text{LINEAR}}(x_i, x_j) = \sum x_i x_j \]  

(4.20)

The basic 4 types of kernel functions for SVM and that are: linear, polynomial, sigmoid and radial basis function among which linear is best for classification task [46]. So for carrying out this research work linear kernel has been used. Default penalty value for linear kernel is chosen by SVMlib is 1. Algorithm 4.2 has been implemented using linear kernel SVM for this experiment.

---

**Algorithm 4.2: SVM Training Algorithm**

**Step 1:** Load the training data set of \( n \) data points, \( \{ x_i, y_i \}, i = 1, \ldots, n \), where \( x_k \) is the \( k \)-th input vector and \( y_k \) is the corresponding \( k \)-th target with values \( \{ -1, +1 \} \).

**Step 2:** Create arbitrary weights for every input data point.

**Step 3:** Determine the value of the bias term \( w_0 \) and initialize the error \( \xi \) for each point randomly.

**Step 4:** Initialize \( C \) using random values.

**Step 5:** Search for values of \( w_0, \xi \), and \( w \) that minimizes the objective function, (4.17) and, (4.18).

**Step 6:** Calculate the number of support vectors \( \alpha \) using, (4.19).
Step 7: Training data for an SVM model could be classified using, (4.19) with linear kernel function, (5).

Step 8: Classify any new point using linear kernel function, (4.20).

Step 9: Loop until stopping criteria is satisfied, usually until reaching the maximum number of iterations.

4.6.7 Experimental Result

For carrying out the experiment for the question classification by using an SVM machine learning approach the operating platform is used here is MATLAB.

For finding the best combinations of lexical, semantic and syntactic features which can give the best accuracy among all other features, the experiment has been performed for the different combinations as like the previous chapter.

The combination of three features, lexical, semantic and syntactic for 1000 questions provides the accuracy 89% for coarse grain for the combination of unigram, stemming, headword, question prototype, named entity recognizer and semantic headword features which are represented in figure 4.29. Similarly, figure 4.30 shows for fine grain accuracy reaches 84.4% when unigram, headword, question prototype, named entity recognizer and semantic headword features are combined.

All the experiment performed above is repeated for increasing the no of question which is shown in figure 4.31 and figure 4.32 for coarse and fine grain respectively and it can observe that the highest accuracy for coarse grain is 96.2% and for fine grain is 91.1% for the setup of 5500 training questions and 500 testing questions. Comparative observation has shown in figure 4.33 for coarse and fine grain classification.
Figure 4.29 Combination of Lexical, Syntactic and Semantic Features and their Accuracy for Coarse Grain for 1000 Questions (using SVM Classifier)
Figure 4.30 Combination of Lexical, Syntactic and Semantic Features and their Accuracy for Fine Grain for 1000 Questions (using SVM Classifier)
Figure 4.31 Accuracy of 5500 Questions for the Combination of Lexical, Semantic and Syntactic Features for Coarse Grain (using SVM Classifier)

Figure 4.32 Accuracy of 5500 Questions for the combination of Lexical, Semantic and Syntactic Features for Fine Grain (using SVM Classifier)
4.6.8 Result Analysis & Discussion

Result Analysis

After the training phase result is obtained by finding the accuracy of the classifier

\[
\text{Accuracy} = \frac{\text{No. of Question Correctly Classified}}{\text{Total Number of Input Question}}
\]

For 1000 questions training and testing are provided for the different combination of features, the result acquired as accuracy 89% for coarse grain classification 84.4% for fine grain classification. In the below mentioned table 4.2 the result for increasing number of questions is represented, for 5500 questions of coarse grain classification the result obtained is 96.2% and for fine grain the result is 91.1%.
Table 4.2 Accuracy of Classification for Increasing no of Questions of Coarse and Fine grain (using SVM Classifier)

<table>
<thead>
<tr>
<th>No. of Questions</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy in Coarse grain</td>
<td>89.0%</td>
<td>93.2%</td>
<td>94.4%</td>
<td>95.8%</td>
<td>96.2%</td>
</tr>
<tr>
<td>Accuracy in Fine grain</td>
<td>84.4%</td>
<td>87.8%</td>
<td>88.8%</td>
<td>89.4%</td>
<td>91.1%</td>
</tr>
</tbody>
</table>

Discussion

Obviously it can be observed that the result obtained by SVM for the question classification accuracy is 96.2% for coarse grain, which is outstanding than the 80.6%, which is obtained from Naïve Bayes classifier and 93.2% which is obtained by BPFFANN. Similarly for fine grain the classification accuracy obtained from BPFFANN is 89.2% and from Naïve Bayes classification accuracy is 78.6%, whereas by using SVM the accuracy acquired is 91.1% which is again much higher than the Naïve Bayes and BPFFANN accuracies. The result obtained from SVM is pretty good and just removing the problem associated with neural network that are: long training time, local minima and over-fitting. SVM is also very good alternative for working with a large dataset.

But the problem associated with SVM is the tuning of penalty parameter C because the linear kernel has used in this work so no need to tune other parameters only the proper value of C is needed. Selection of the appropriate value of C is very important because the classification accuracy depends on the value of C. There is a concern of C along with the complexity and percentage of non-separable. The penalty parameter provides an alert for the SVM optimization that, how really it would like to keep away training examples from mis-classification and generally it is selected by the user.

On the other hand, the proper selection of the penalty parameter C for each regularization method is extremely significant. If the value that is selected for C is extremely large, it has a high penalty for non-separable points and have to store many support vectors and over-fit. There will be the difficulty of under fitting if the value of C is too small, it can also increase the number of training errors, while a large value of C will lead to a behavior similar to that of a hard-margin SVM.
Alternatively, an extremely small value of C will cause the optimizer to look for a larger-margin for dividing hyper plane, although that hyper plane mis-classifies more points and should get mis-classified examples, often even if our training data is linearly separable.

So for further improvement, the research work proposes Particle Swarm Optimization (PSO) technique for optimizing the penalty parameter C in linear kernel for SVM in a subsequent chapter and also it will give the significant improvement in the classification technique and also will improve the classification accuracy.

4.6.9 Summary

The present section deals with discussion about the result that has been obtained from Support vector machine classifier using the linear kernel for question classification. The algorithm has been presented for question classification by using SVM with the combinations of various features extracted from question set.

<table>
<thead>
<tr>
<th>Table 4.3 Relative Comparison of the Results for Different Classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No.of Questions</strong></td>
</tr>
<tr>
<td><strong>Naïve Bayes Classifier Accuracy</strong></td>
</tr>
<tr>
<td>Coarse Grain</td>
</tr>
<tr>
<td>Fine Grain</td>
</tr>
<tr>
<td><strong>BPFFANN Accuracy</strong></td>
</tr>
<tr>
<td>Coarse Grain</td>
</tr>
<tr>
<td>Fine Grain</td>
</tr>
<tr>
<td><strong>SVM Classifier</strong></td>
</tr>
<tr>
<td>Coarse Grain</td>
</tr>
<tr>
<td>Fine Grain</td>
</tr>
</tbody>
</table>

The comparative analysis of the result for different classifier is presented in table 4.3. The study and outcome have been observed for naive bayes, BPFFANN and support vector machine. The output that has been obtained from support vector machine classifier by using linear kernel is outstanding as compared to naive bayes classifier and the Back-Propagation-Feed Forward-Artificial- Neural-Network classifier.