Implementation of Intelligent Feature in Bangla

Sarifunnahar[1], Mohammad Shamsul Arefin[1] and Md. Raju Ahmed[2]
[1]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Bangladesh
[2]Department of Electronic & Telecommunication Engineering, International Islamic University Chittagong
E-mail: sarifun02csecuet@yahoo.com, sarefin@cuet.ac.bd and mdrazu@ete.iiuc.ac.bd

Abstract: *Intelligent is an auto-completion-like feature presented in many popular Integrated Development Environment (IDE) & editors. It makes the user more comfortable and offers enjoyable experience using the related product. But there is no implementation in Bangla of this nice feature. The emergence of the Unicode standard and the availability of tools supporting it are among the most significant recent global software technology trends. The main objective of this paper is to include intelligent feature in Bangla, which will create a new dimension in the field of statistical information processing.*

Key Words: NLP, statistical analysis, word sense disambiguation

1. INTRODUCTION

The task of predicting the most likely word based on properties of its surrounding context is the archetypical prediction problem in Natural Language Processing (NLP). In many NLP tasks, it is necessary to determine the most likely word, part-of-speech (POS) tag or any other token, given its history or context. Examples include part-of speech tagging, word-sense disambiguation, speech recognition, accent restoration, and word choice selection in machine translation, context-sensitive spelling correction and identifying discourse markers. Most approaches to these problems are based on n-gram-like modeling. Namely, the learning methods make use of features, which are conjunctions of typically three consecutive words or POS tags in order to derive the predictor.

The most influential problem in motivating statistical learning application in NLP tasks is that of word selection in speech recognition by F. Jelinek et al. [1]. There, word classifiers are derived from a probabilistic language model, which estimates the probability of a sentence using Baye's rule as the product of conditional probabilities.

Machine learning based classifiers and maximum entropy models, which, in principle, are not restricted to features of these forms, have used them nevertheless, perhaps under the influence of probabilistic methods by E. Brill et al. [2]. It has been argued that the information available in the local context of each word should be augmented by global sentence generative probabilistic model, typically using Markov or other independence assumptions, which gives rise to estimating conditional probabilities of n-grams type features.

Computer implementation of intelligent feature in Bangla is still limited. Working with Bangla in an active research area. Some features have been developed like, Bangla to English converter, which takes Bangla words as input & produce English words as output. S. A. Hauladar et al. [3] has performed this work. Another feature, English to Bangla converter, which takes English words as input & produce Bangla words as output. M. Kamruzzaman [4] has performed this work. Bangla text encryption & decryption using RSA algorithm has performed by M. F. Islam [5]. M. Islam [6] has performed the work. of Bangla numerical recognition, which will automatically generate by using fuzzy linguistic rule in online handwriting system.

But prediction related work has not yet been developed. In this paper we have proposed such a method for Bangla documents.

2. LITERATURE REVIEW

2.1 Introduction of Statistical Inference Analysis

Statistical NLP aims to do statistical inference for the field of natural language. Statistical inference in general consists of taking some data and then making some inferences about this distribution. For example, we might look at lots of instances of prepositional phrase attachments in a corpus and use them to try to predict prepositional phrase attachments for English in general. The discussion in this chapter divides the problem into three areas: dividing the training data into equivalence classes, finding a good statistical estimator for each equivalence classes, and combining multiple estimators..

2.2 Reliability versus Discrimination

In order to do inference about one feature, we wish to find other features of the model that predict it. Here, we are assuming that past behavior is a good guide to what will happened in future. This gives us a classification tasks: we try to predict the target feature on the basis of various classificatory features.

2.2.1 n-gram models

The task of predicting the next word can be stated as attempting to estimate the probability function P:

P (wn, w1,……wn-1) ----------(2.1)

In such a stochastic problem, we use a classification of the previous words, the history, to predict the next

word. The cases of n-gram models that people usually use are for n=2, 3, 4, and these alternatives are usually referred to as a bigram, a trigram and a four gram model, respectively.

2.3 Statistical Estimators

Given a certain number of pieces of training data that fall into a certain bin, the second goal is then finding out how to derive a good probability estimate for the target feature based on these data. For our running example of n-grams, we will be interested in P (w1, w2, …….., wn) and the prediction task P (wn / w1……..wn-1). Since: P (wn/w1……..wn-1) = P (w1, w2,……..wn)/ P (w1……..wn-1)- (2.2)
Estimating good conditional probability distribution can be reduce to having good solutions to simply estimating the unknown probability distribution of n-grams.

2.3.1 Laplace's Law

The manifest failure of maximum likelihood estimation forces us to examine better estimators. The oldest solution is to employ Laplace's law. According to this law,
PLap (w1……..wn) =C (w1…….wn.)+1/ (N+B) ----------(2.3)
This process is often formally referred to as adding one and has the effect of giving a little bit of the probability space to unseen events.

2.4 Combining Estimator

In this section, we consider the more general problem of how to combine multiple probability estimates from various different models. If we have several models of how the history predicts what comes next, then we might wish to combine them in the hope of producing an even better model.

2.4.1 Simple Linear Interpolation

One way of solving the sparseness in a trigram model is to mix that model with bigram and unigram models that suffer less from data sparseness. For interpolating n-gram language models, such as deleted interpolation from a trigram model, the most way to do this is:
Pli (wn / wn-2, wn-1) = λ1P1 (wn) + λ2P2 (wn / wn-1) λ3P3 (wn / wn-1, wn-2)----------(2.4)
Where 0<= λi<=1.

2.4.2 General Linear Interpolation

In simple linear interpolation, the weights are just single number, but one can define a more general and powerful model where the weights are a function of history. For K probability functions Pk the general form for a linear interpolation model is:
Pli (w / h) =∑ λi (h) Pi (w / h) ----------(2.5)
Where 0<= λi (h) <=1, 1<=i<=k

Linear interpolation is commonly used because it is a very general way to combine models. Randomly adding in dubious models to a linear interpolation need not do harm providing one finds a good weighting of the models using the EM algorithm. But linear interpolation can make bad use of component models, especially if there is not a careful partitioning of the histories with different weights used for different sorts of histories.
A number of smoothing methods are available which often offer similar and good performance figures. Using good turing estimation and linear interpolation or back off to circumvent the problems of sparse data represent good current practice.

2.5 Proposed Work

The work presented in this paper is directly related to statistical natural language processing (auto completion system). An NLP system needs to determine something of the structure of text at least enough that it can answer. Practical NLP system must be good at making disambiguation decision of word sense, word category, syntactic structure and semantic scope. As a running example of statistical estimation, we will examine the classic task of language modeling, where the problems are to predict the next word given the previous word, fill in the blanks, words frequency, etc.
This auto completion task is fundamental to speech or optical character recognition and is also used for spelling correction, handwritten recognition and statistical machine translation. Moreover, this feature not only reduces the human effort in producing NLP system but also interesting scientific issues regarding human language acquisition.

3. METHODOLOGY

3.1 Bangla Auto Completion Feature for Primarily Three Keywords

First part of this work is to predict the fourth word of a sentence. After primarily three keywords, the fourth word will be selected depending on word counts.
At first, we will store a Bengali document in a word file. This file capacity is unlimited.
Then we take Bangla text from file as input. Every time we press a key, the key is checked if it is 'space' or not. If it is the third time space then auto-completion feature is activated and the list-box is made visible. Now the list-box will get focus which will contain fourth words of all the sentences and at the same time, every word should appear with its counting number (i.e. if the word appears more than once or not then counting in bracket). Again key-press is monitored for navigation keys. Then the highest number of word is taken by matching the third word and the list-box is removed. If two or

more words have the same counting number, anyone can be selected randomly.
The flowchart of this process is given in Figure 1. Here we see that when input in Bangla is taken text from file, key is pressed in the Rich Edit box and send the control string to Parser. If parser gets the word as a keyword, then auto completion feature is activated. Then generate list box with appropriate words from dictionary based on next word count.

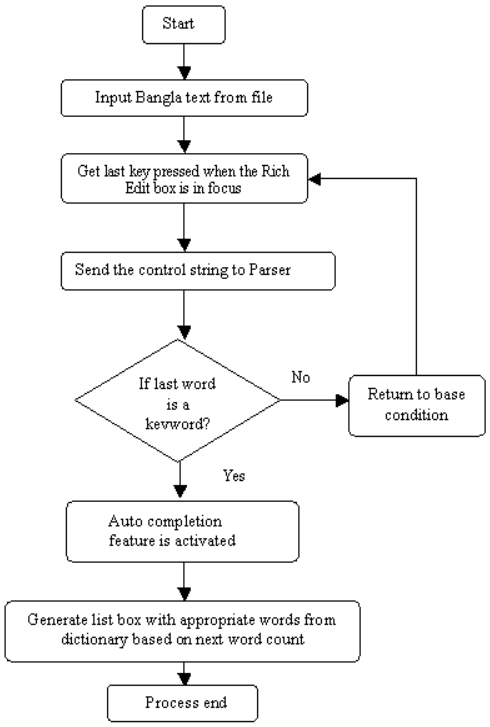

Figure 1: Bengali auto completion feature for primarily three keywords

3.2 Bangla Auto Completion Feature for Every Word

Second part of this project work is to make variable of prediction system, i.e. auto completion will activated one after another word in a sentence.
For that we will store a Bengali document in a word file. This file capacity is unlimited. Then we take Bangla text from file as input. First we press a key. The key is checked if it is 'space' or not. If it is the first time space then auto-completion feature is activated and the list-box is made visible. Now the list-box will get focus which will contain second words of all the sentences and at the same time, every word should appear with its counting number (i.e. if the word appears more than once or not then counting in bracket). Then the highest number of word is taken by matching the first word and the list-box is removed. If two or more words have the same counting number, anyone can be selected randomly. Again key-press is monitored & if it get 'space' then auto-completion feature is activated and the list-box is made visible, which will contain third words of all the sentences and at the same time, every word should appear with its counting number. Then selecting word as previous process and this auto completion feature will continue until the end of line or sentence. The flowchart of this process is given in Figure 2.

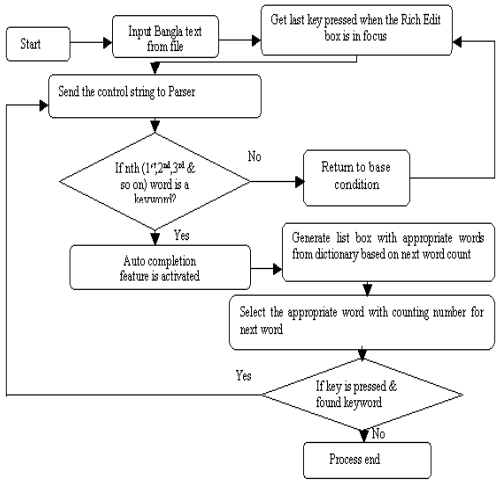

Figure 2: Bengali auto completion feature for every word

3.3 Fill in the Blanks for Primarily Three Keywords

Third part of this project work is to fill in the blanks after primarily three same keywords. All fourth word should be selected depending on word counts.
For that we will store some Bengali sentences in a word file. First three keywords are same of all sentences but fifth words are sometimes same and sometimes are not. This file capacity is unlimited. Then we take Bangla text from file as input. Every time we press a key, the key is checked if it is 'space' or not. If it is the third time space then we use dashed mark ('---') to represent the fourth word. Again fourth time space is required and key is pressed to complete fifth word. After completing fifth word, we use a dot mark ('.') to activate auto-completion feature. When key is pressed as a dot mark, auto-completion is feature activated in dashed mark and the list-box is made visible. Now the list-box will get focus which will contain those fourth words of sentences which matching with third and forth words and at the same time, every word should appear with its counting number (i.e. if the word appears more than once or not then counting in bracket). Again key-press is monitored for navigation keys. Then the highest number of word is taken and the list-box is removed. If two or more words have the same counting number, anyone can be selected randomly. The corresponding flowchart of this process is given in Figure 3.

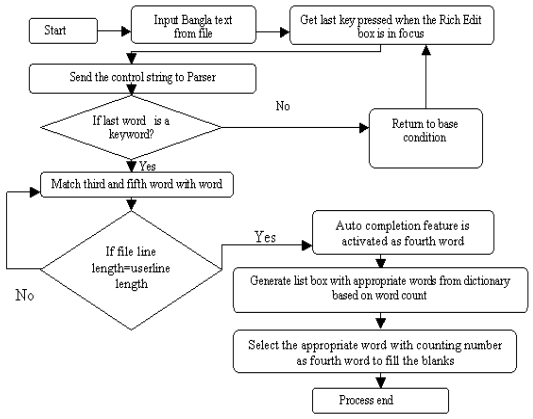

Figure 3: Fill in the blanks for primarily three keywords

4. EXPERIMENTAL RESULTS AND DISCUSSION

For every step of work here we need a dictionary file or resource file, which will store a Bengali document. This file capacity is unlimited.

Such a resource file is graphically shown in Figure 4.

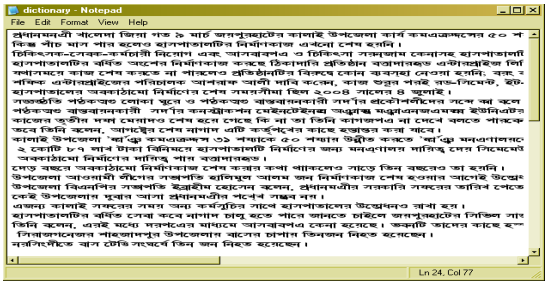

Figure 4: Dictionary file

4.2 Bengali Auto Completion Feature for Primarily Three Words

First part of this project work is to predict the fourth word of a sentence with word count.

When we run this program after compilation we see a window, which contains fourth words of each sentences with counting number and this is given in Figure 5.

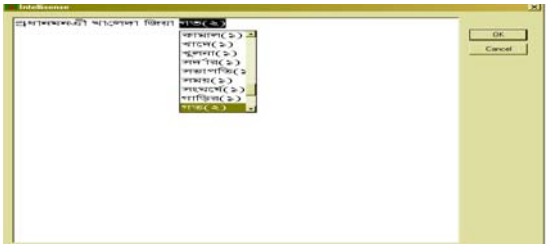

Figure 5: Frequencies of fourth words after first three words of documents in Figure 4.

4.3 Bangla Auto Completion Feature for Every Word

According the same source file of Figure 4, second step of the work is to make variable of prediction system, i.e. auto completion will activated one after another word in a sentence. After running the program frequencies of second words after first word of documents in Figure 4 is shown in figure 6.

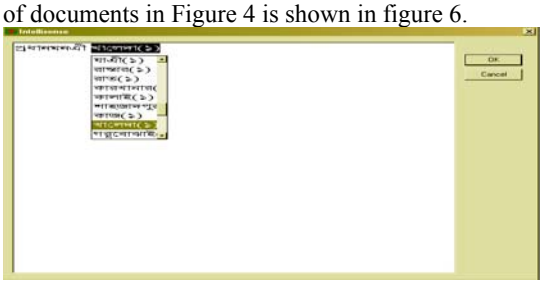

Figure 6: frequencies of second words after first word of documents in Figure 4

4.4 Fill in the Blanks for Primarily Three Keywords

Third step of this project work is to fill in the blanks after primarily three same keywords with counting number. All fourth word should be selected depending on word counts. We need a resource file which contains sentences is given in Figure 7.

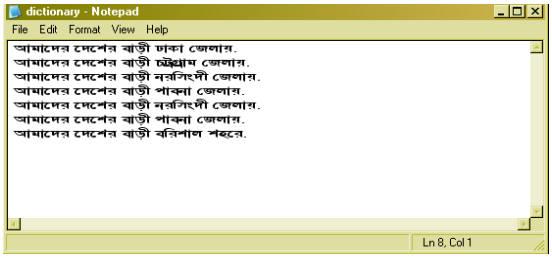

Figure 7: Dictionary file

The answer can be found from after running the program and this is given in Figure 8.

Figure 8: Output of filling the gaps using statistical analysis

4.5 Graphical Representation of Words Versus Frequency

Fourth and last part of this project work is graphical representation of words versus frequency. These words are the fourth words of sentences and counting numbers of fourth words are represented according to their frequencies. The input file is given in Figure 9 and the corresponding result of this statistical analysis is given in Figure 10.

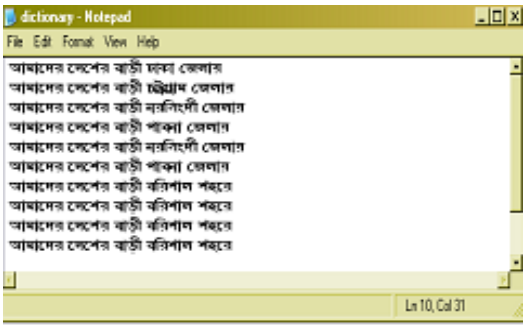

Figure 9: Dictionary file

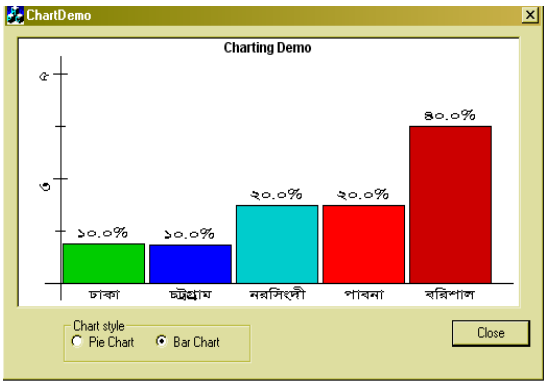

Figure 10: Corresponding bar chart

5. CONCLUSION

In this work, we have tried to implement intelligent feature on Bengali text documents. Here we have generated an approach to word prediction that is based on learning a representation for each word as a function of previous keywords and linguistics predicates in its context. The eventual goal of a language model is to accurately predict the value of a missing word given its context. Given a prediction task (a sentence with a missing word) the word representations are evaluated on it and complete for the most likely word to complete the sentence. Here, we have done this task after primarily 3 keywords i.e. the missing word is the fourth word of the sentences. Last of all, we have shown graphical representation of words versus frequency.

The results of this project shows that intelligent feature can be implemented in Bangla sentences and this feature can be used to fill the blanks by accurately predicting words matching with the other words of sentences. This project work has a nice feature of word count. By graphically, this project also gives an accurate result to represent words frequency.

This project has some limitations. In fill in the blanks, only the missing word can be generated after primarily three keywords but not for all. In graphical representation of words versus frequency, the words are all fourth time words but not for all words.

In this project work, there is a portion to make variable of prediction system, i.e. auto completion feature will activated one after another word in a sentence. Here list boxes are generated which contains all of the words of all the sentences with counting number. The next work may be to create list boxes, which contain only probabilistic matching words with the previous word sense.

In another portion, fill in the blanks after primarily 3 same keywords i.e. the missing word is the fourth word of the sentence. All fourth word should be selected depending on word counts. The next work may be fill in the blanks after every word i.e. the missing word can be any word of the sentence. Another future work is to find out ‘true’ or ‘false’ by matching with the source file sentences.

References

[1] C. Chelba and F. Jelinek “Exploiting syntactic
structure for language modeling”, In COLING-ACL, 1998.

[2] E. Brill. “Transformation-based error-driven learning and natural language processing”, A case study in part of speech tagging. Computational Linguistics, 1995.

[3] S.A. Haoladar “Bangla to English converter”, undergraduate thesis, Dept of CSE, Chittagong University of Engineering & Technology, 2002.

[4] M. Islam “Online hand written Bengali Numerical Recognition with Automatically Generated Fuzzy Linguistic Rule”, undergraduate thesis, Dept of CSE, CUET, 2005.

[5] M. Kamruzzaman, ”English to Bangla converter”,
undergraduate thesis, Dept of CSE, Chittagong University of Engineering & Technology, 2006.

[6] M. F. Islam”Bangla text encryption & decryption
using RSA algorithm”, undergraduate thesis, Dept. of CSE, Chittagong University of Engineering & Technology, 2006.

[7] C. D. Manning, H. Schutge “Foundations of Statistical Natural language Processing”.