An Investigative Design Based Statistical Approach for Determining Bangla Sentence Validity

Md. Riazur Rahman†, Md. Tarek Habib†, Md. Sadekur Rahman†, Shaon Bhatta Shuvo†, Mohammad Shorif Uddin††

†Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
††Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh

Summary
Automatic grammatical verification of sentences is an essential task in natural language processing. There has been a scarcity of resources in Bangla for such tasks. To address this issue this paper presents a new n-gram based statistical approach to check the syntactic and semantic correctness of sentences in Bangla. An n-gram frequency count-based probabilistic language model is employed combining standard n-gram statistics with appropriate smoothing and advanced backoff language model to detect validity of any sentence in Bangla to design the proposed method. A new Bangla corpus of 10 million words is used to train the proposed method. The system was tested on both valid and invalid sentences collected separately from training corpus. In terms of detecting correct and incorrect sentences the proposed system achieved 82% precision and 81% recall scores outperforming the existing systems.

Key words:
Sentence validity detection; natural language processing; n-gram; smoothing; backoff strategy; language model..

1. Introduction

Identifying the grammatical correctness of a sentence is an emerging research area in natural language processing (NLP). Checking the validity of a text is the task of determining whether the text in question is proper with respect to the grammatical regulations of the respective language. An automated system that can judge the correctness of a given text is very useful in many application such as word processors, compilers, text messaging system, computer aided language learning systems etc.

There are three methodologies that are widely used for the purpose of grammar checking namely syntax-based checking, rule-based checking and statistics-based checking. Syntax-based method [1] works by building a parse tree or table for each given sentence. The sentence is deemed valid if the parsing process succeeds. Otherwise the sentence is marked as invalid. In case of rule-based method [2], a set of manually developed grammatical rules are used to determine to correctness of the given text. On the contrary, in case of statistics-based approach [3], a statistical language model (SLM) is built from a text corpus of the target language that can estimate the distribution of the language as accurately as possible. A SLM is a probability distribution $P(s)$ over strings S that attempts to reflect how frequently a string S occurs as a sentence. The target text is regarded as invalid if the SLM probability score for it is below some threshold. Though there are a lot of tools & techniques developed for grammar checking in recent years such as Grammarly, WhiteSmoke, CorrectEnglishComplete etc. [4], there is still a lot of scope to improve the performance of grammar checking systems.

In the last few years a lot effort has been made on the detection of correct and incorrect sentences in many different languages such as English, French, and Chinese etc. [5]. Whereas although Bangla ranks 7 among the most spoken languages in the world with more than 250 million native speakers, surprisingly there is a large scarcity of resources for Bangla sentence error detection [6, 7].

To address the above mentioned issues in this work, a new statistical method is proposed which used n-gram based language model combined with Witten-Bell smoothing and Backoff language modeling strategy [8, 9] to decide the validity of a sentence in Bangla. The presented technique was trained on a large Bangla corpus of 10 million words collected from various sources such as online newspapers, blogs, literature etc. A strategy was developed to determine appropriate threshold to distinguish between valid and invalid sentences. The threshold was finalized by performing a 5-fold cross validation [9] on the training set. The proposed method was tested on a test set of 10000 valid and 10000 invalid sentences. The proposed method outperforms the existing systems achieving 82% precision and 81% recall on the test set.

The rest of the paper is organized as follows; section 2 presents a review of the previous works on Bangla grammar checking while some theoretical background on

Manuscript received November 5, 2016
Manuscript revised November 20, 2016

n-gram based sentence probability calculation is provided in section 3. Whereas section 4 describes the methodology used for developing the system. Section 5 presents the experimental results while section 6 concludes the paper.

2. Related works

There has been very little development in grammar checking for Bangla language. The authors in [10] proposed a context free grammar based predictive parser to recognize grammaticality of Bangla sentences. On the other hand, in [11] the authors presented an n-gram based statistical grammar checking system for Bangla, which used the n-gram frequency based probability analysis of parts-of-speech (POS) tags of words to decide whether the sentence is grammatically correct or not. Their method suffers from the zero frequency problem [8] which severely degrades the performance of the system. Also the fact that they only used POS tag information made their method only useful for detecting syntactic structure of the sentence missing on the semantic information. They used a very small corpus of only 5000 words to build the n-gram model. Using this model they reported a moderate success rate for only detecting correct sentences on a very tiny test set of 378 sentences. In a recent work [12], another n-gram based statistical method was proposed. In this work, rather than using frequency of POS tags of words the authors used n-gram frequency based probability analysis of words to train and test their system. To resolve the zero frequency problem of n-gram models, they used Witten-Bell discounting [8] with their n-gram model. They trained their statistical n-gram model with a small experimental corpus of 1 million words with a test set of 1000 correct and 1000 incorrect sentences. But in their approach, the authors used a manually selected predefined threshold to separate the valid and invalid sentences which is not a practical approach if the method is trained and tested in different data sets.

As mentioned above there is clearly no comprehensive and reliable grammar checker available for Bangla yet. This motivated us to develop a robust sentence grammaticality detection method for Bangla language.

3. N-gram based Sentence Probability Calculation

3.1 N-Grams

N-grams [13] of texts are extensively used in text mining and natural language processing tasks. An n-gram is the pair of a word sequence $w_{i-n+1}...w_i$ containing n words and its according count, based on the occurrences of the sequence in a corpus. More concisely, an n-gram model predicts the probability of a word w_i based on the probability of $w_{i-n+1}...w_{i-1}$ words sequence. According to probability theory, this can be written as $P(w_i \mid w_{i-n+1}...w_{i-1})$. When used for language modeling, independence assumptions also known as markov assumptions [8] are made so that each word depends only on the last n-1 words. This probability can be calculated as,

$$P(w_i \mid w_{i-n+1}...w_{i-1}) = \frac{C(w_{i-n+1}...w_i)}{\sum_w C(w_{i-n+1}...w_{i-1}w)} \quad (1)$$

$C(w_{i-n+1}...w_i)$ is the count of occurrences of word sequence $w_{i-n+1}...w_i$ and $\sum_w C(w_{i-n+1}...w_{i-1}w)$ indicates the sum of counts of all the n-grams that starts with $w_{i-n+1}...w_{i-1}$. When n = 1 it is called unigram or 1-gram. For n = 2, it is known as bigram or 2-gram. N-grams with n = 3 is called trigrams. With n = 4, they are termed as quadrigrams or 4-grams and all the higher n-grams are simply termed as n-grams such as for n = 5 the model is known as 5-grams.

3.2 Sentence Probability Calculation using N-grams

To calculate the probability of a sentence using an n-grams language model, first the probability of each possible n-grams of words in the sentence are calculated. Then these n-gram probabilities are multiplied to find the sentence probability. The higher the probability of a sentence the higher chances it has to be a properly formed sentence of the target language. For a sentence $S = (w_1\ w_2\ w_3 \ldots w_N)$ with N words separated by blank space, the probability of S can be computed as below,

$$P(S) = \prod_{i=1}^{N} P(w_i \mid w_{i-n+1}...w_{i-1}) \quad (2)$$

For example, the probability calculations of the sentence "সুবিধাবঞ্চিত শিশুদের বইগুলো দেওয়া হবে" by using unigram, bigram, trigram, quadgram models are shown below:

Unigram Probability,

P("সুবিধাবঞ্চিত শিশুদের বইগুলো দেওয়া হবে") = P(সুবিধাবঞ্চিত)* P(শিশুদের)* P(বইগুলো)* P(দেওয়া)* P(হবে)

Bigram Probability,

P("সুবিধাবঞ্চিত শিশুদের বইগুলো দেওয়া হবে") = P(সুবিধাবঞ্চিত|<s>)* P(শিশুদের | সুবিধাবঞ্চিত)* P(বইগুলো | শিশুদের)* * P(দেওয়া | বইগুলো)* P(হবে | দেওয়া)*P(</s> | হবে)

Trigram Probability,

P("সুবিধাবঞ্চিত শিশুদের বইগুলো দেওয়া হবে") = P(সুবিধাবঞ্চিত | <s> <s>)* P(শিশুদের | <s> সুবিধাবঞ্চিত)* P(বইগুলো | সুবিধাবঞ্চিত শিশুদের)* P(দেওয়া | বইগুলো শিশুদের)* P(হবে | শিশুদের দেওয়া)* P(</s> | দেওয়া হবে)

Quadgram Probability,

P("সুবিধাবঞ্চিত শিশুদের বইগুলো দেওয়া হবে") = P(সুবিধাবঞ্চিত | <s> <s> <s>)* P(শিশুদের | <s> <s> সুবিধাবঞ্চিত)* P(বইগুলো | <s> সুবিধাবঞ্চিত শিশুদের)* P(দেওয়া | সুবিধাবঞ্চিত বইগুলো শিশুদের)* P(হবে | বইগুলো শিশুদের দেওয়া)* P(</s> | শিশুদের দেওয়া হবে)

These probabilities are normalized to be within the range of 0 to 1.

3.3 Zero Frequency Problem & Discounting

No matter how large a training corpus is, it cannot cover a natural language entirely. There will always be some perfectly acceptable word sequences that are missing from the corpus. This means, there will be many cases of acceptable zero probability n-grams that should have some non-zero probability.

Consider the words that follow the bigram "দেশের বৃহত্তম" in our corpus with their counts:

দেশের বৃহত্তম পর্বতমালা: 5

দেশের বৃহত্তম শপিংমল: 3

দেশের বৃহত্তম জেলা: 5

But suppose our test set contains texts such as followings:

দেশের বৃহত্তম কারখানা

দেশের বৃহত্তম খানা

The n-gram model will incorrectly evaluate the probability P(খানা | দেশের বৃহত্তম) as 0.

These zero frequency words that never occur in training set but occur in the test sets poses serious problems. Firstly because they indicate the underestimation of all sorts of word sequences that may appear. Secondly, since the probability of a sentence is calculated by multiplying the n-grams of different word sequences if any of the n-gram has a zero probability, the entire sentence will have zero probability which will miss-calculate the correct sentences with zero probability.

To keep a language model from assigning zero probabilities to unseen words or contexts, a small portion of probability mass is taken from the more frequent words or word sequences and distributed to unseen events i.e. in this case unknown words or contexts. This process is known as discounting. There are several discounting algorithms available such as Add-One discounting, Witten-Bell discounting, Good-Turing discounting etc. [10, 12]. The Witten-Bell discounting is chosen in this work due to its simplicity and robustness. Witten-bell discounting technique will be discussed in details next.

3.4 Witten-Bell Discounting

Witten-Bell (WB) discounting uses the counts of events occurring at least once to estimate the counts of events that never occurred. To compute the counts of all n-grams that has been seen in the corpus at least once, one need to calculate the number of n-gram types since each unique n-gram is present at least once in the training corpus. WB discounting works by taking some of the probability mass from n-grams that are seen at least once to distribute them among the n-grams that are never seen in the training data to prevent any n-gram from having zero probability. The total probability mass that is discounted to all the zero n-grams is calculated as below:

$$\delta(w_{i-n+1}\cdots w_{i-1}) = \frac{T(w_{i-n+1}\cdots w_{i-1})}{T(w_{i-n+1}\cdots w_{i-1}) + N(w_{i-n+1}\cdots w_{i-1})} \quad (3)$$

$\delta(w_{i-n+1}\cdots w_{i-1})$ indicates the total probability mass discounted for zero n-grams with the context $w_{i-n+1}\cdots w_{i-1}$. $T(w_{i-n+1}\cdots w_{i-1})$ is the number of n-gram types with a common preceding word sequence of $w_{i-n+1}\cdots w_{i-1}$ and $N(w_{i-n+1}\cdots w_{i-1})$ represents the total number of n-gram tokens that starts with $w_{i-n+1}\cdots w_{i-1}$ context. If $Z(w_{i-n+1}\cdots w_{i-1})$ indicates the total number of zero n-grams starting with history $w_{i-n+1}\cdots w_{i-1}$, then the probability for any zero count n-gram can be easily calculated as,

$$P^{WB}_{c(w_{i-n+1}\cdots w_i)=0}(w_i \mid w_{i-n+1}\cdots w_{i-1}) = \frac{T(w_{i-n+1}\cdots w_{i-1})}{Z(w_{i-n+1}\cdots w_{i-1}) * \{T(w_{i-n+1}\cdots w_{i-1}) + N(w_{i-n+1}\cdots w_{i-1})\}} \quad (4)$$

Since the total probability mass must equal to 1, the leftover probability mass of for all non-zero count n-grams can easily be calculated as follows:

$$1 - \delta(w_{i-n+1}\cdots w_{i-1}) = \frac{N(w_{i-n+1}\cdots w_{i-1})}{T(w_{i-n+1}\cdots w_{i-1}) + N(w_{i-n+1}\cdots w_{i-1})} \quad (5)$$

Now the probability for any n-gram with non-zero count can be computed as,

$$P^{WB}_{c(w_{i-n+1}\cdots w_i)>0}(w_i \mid w_{i-n+1}\cdots w_{i-1}) = \frac{N(w_{i-n+1}\cdots w_{i-1}) * C(w_{i-n+1}\cdots w_i)}{T(w_{i-n+1}\cdots w_{i-1}) + N(w_{i-n+1}\cdots w_{i-1})} \quad (6)$$

Backoff language model is discussed next.

3.5 Backoff N-gram Language Model

Introduced by Katz in 1987, Backoff (BO) language model [14] for n-grams is a non-linear method that builds an n-gram language model based on an (n-1)-gram model. BO model works on the principle that if a higher order n-gram has non-zero count then it only uses the higher order counts to calculate the probability. But if the higher order n-gram has zero count, then it backs off to the lower order n-gram i.e (n-1)-gram model to calculate the probability. The general form of recursive BO model is written below,

$$P^{BO}(w_i \mid w_{i-n+1}\cdots w_{i-1}) = \begin{cases} P^*(w_i \mid w_{i-n+1}\cdots w_{i-1}), & \text{if } C(w_{i-n+1}\cdots w_i)>0 \\ \alpha(w_{i-n+1}\cdots w_{i-1})P^{BO}(w_i \mid w_{i-n+1}\cdots w_{i-1}), & \text{otherwise} \end{cases} \quad (7)$$

Since in BO model, algorithm backs off to lower order model when the probability of n-gram is zero, extra probability mass gets added into the equation making the total probability of an n-gram or word sequence greater than 1 which is undesirable. So, in BO language models some probability mass needs to be discounted from the higher order models to lower order models. In (7), $P^*(w_i \mid w_{i-n+1}\cdots w_{i-1})$ is the discounted probability and $\alpha(w_{i-n+1}\cdots w_{i-1})$ is the backoff weight that denotes the amount of probability mass discounted to an (n-1)-gram. The discounted probabilities are calculated using Good-Turing [15] estimates as follows,

$$P^*(w_i \mid w_{i-n+1}\cdots w_{i-1}) = \frac{C^*(w_{i-n+1}\cdots w_i)}{\Sigma_w C(w_{i-n+1}\cdots w_{i-1}w)} \quad (8)$$

Where $C^* = C^*(w_{i-n+1}\cdots w_i)$ is calculated as,

$$C^* = (C+1)\frac{N_{C+1}}{N_C} \quad (9)$$

In (9), N_C is the number of n-grams with count C.

The backoff weight $\alpha(w_{i-n+1}\cdots w_{i-1})$ is computed as follows,

$$\alpha(w_{i-n+1}\cdots w_{i-1}) = \frac{1-\sum_{w_i:C(w_{i-n+1}\cdots w_i)>0} P^*(w_i \mid w_{i-n+1}\cdots w_{i-1})}{1-\sum_{w_i:C(w_{i-n+1}\cdots w_i)>0} P^*(w_i \mid w_{i-n+2}\cdots w_{i-1})} \quad (10)$$

4. Proposed Method

This work developed an n-gram based statistical language model (LM) combining Witten-Bell (WB) discounting with Backoff (BO) language model strategy to detect syntactic and semantic validity of any sentence in Bangla. The proposed LM is named Witten-Bell Backoff (WBB). The general workflow of the proposed system is depicted in Fig. 1. The proposed method and its related algorithms are discussed in details in the following subsections.

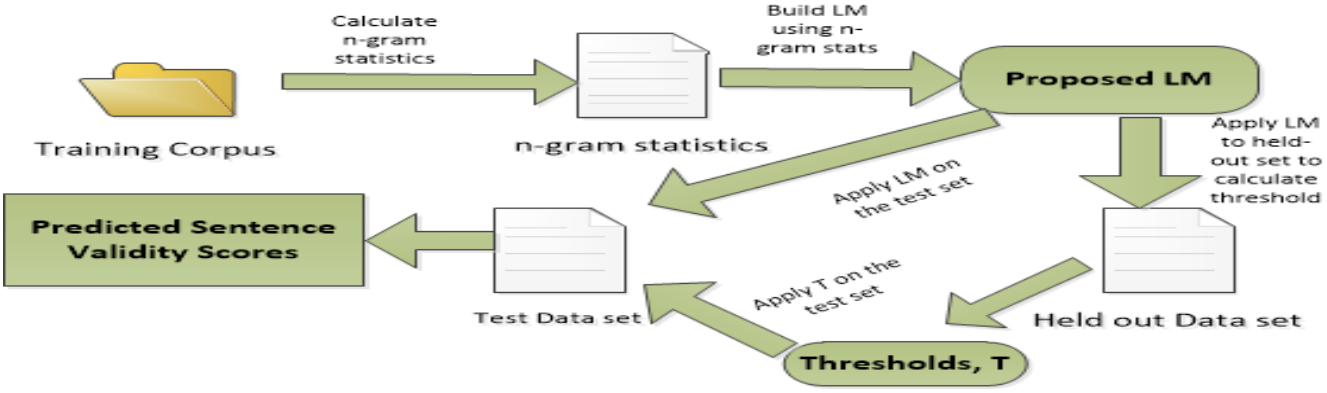

Fig 1. Work Flow Diagram for the Proposed Method

4.1 Proposed Witten-Bell Backoff Language Model

Witten-Bell Backoff (WBB) is a LM which associates the Witten-Bell (WB) discouting into the Backoff (Bo) LM. The general form of WBB LM for computing the probability of an n-gram is as follows,

$$P^{WBB}(w_i \mid w_{i-n+1}\cdots w_{i-1}) = \begin{cases} (1-\delta(w_{i-n+1}\cdots w_{i-1}))P(w_i \mid w_{i-n+1}\cdots w_{i-1}), & \text{if } C(w_{i-n+1}\cdots w_i)>0 \\ \delta(w_{i-n+1}\cdots w_{i-1})P^{WBB}(w_i \mid w_{i-n+1}\cdots w_{i-1}), & \text{otherwise} \end{cases} \quad (11)$$

Where $P(w_i \mid w_{i-n+1}\cdots w_{i-1})$ is the maximum likelihood estimate (MLE) probability of an n-gram defined in (1). The backoff weight for the lower order models in WBB is the discounted probability mass $\delta(w_{i-n+1}\cdots w_{i-1})$ defined in (3). The leftover probability mass $(1-\delta(w_{i-n+1}\cdots w_{i-1}))$ after discounting is defined in (5), which is used to recalculate the discounted probability for the higher order n-grams.

4.2 Training the LMs

To train the language models, a corpus of 10 million word tokens collected online with topics ranging in politics, literature, science, education, sports, music and other news wire was used. The steps for training a LM are listed in Algorithm 1.

ALGORITHM 1. aLGORRITHM for Training LM

1. Extract the sentences from the corpus.
2. Compute and store the n-gram frequencies into the backup storage for n = 1 to 4.
3. Compute the probabilities and backoff weights (if any) for all n-grams calculated in step 2 using appropriate LM and store them in the storage in arpa format.

In this work, WB, BO and WBB all three models were trained for evaluation purpose.

4.3 Testing the LMs

To test if a sentence is valid or invalid; the counts of all the n-grams are first calculated. These frequencies are then used to calculate the probabilities of the n-grams using respective LM methods and training data. The sentence probability score is calculated using (2). If the sentence score is higher than some threshold, it is regarded as valid otherwise invalid. The threshold calculation is discussed in the next section. The detail procedure for testing a list of sentences for validity is presented in Algorithm 2.

ALGORITHM 2. ALGORITHM for Testing Sentences

1. Extract the sentences to be tested from test file.
2. For each sentence *S* Do,
3. Compute *N*, the number of n-grams in *S*.
4. Get the probabilities for the n-grams the using equations (4) & (6) or (7) or (11) for respective LM.
5. Set *score* = 1.
6. For *i* = 1 to *N* Do,
7. Get *p* = the probability of i[th] n-gram.
8. *score* = *score* * *p* .
9. End For
10. If *score* > *T* Do,
11. Predict *S* as valid.
12. Else,
13. Predict *S* as invalid.
14. End For
15. Store the prediction results.

4.4 Threshold Calculation

Language modelling methods to grammatical correctness detection are typically based on a probability score produced by a language model (LM) learned from a large corpus of correct sentences. A valid sentence will usually have higher probability score than an invalid one. With this simple assumption, an initial threshold T_{min} is defined as the minimum probability score of all valid sentences when testing the LM on a held-out data set. To reduce the generalization error and to achieve better performance on the test set, 5-fold cross validation is used in this work. The threshold selection procedure is depicted in Algorithm 3.

ALGORITHM 3. ALGORRITHM for Training LM

1. Given the training corpus *D;* divide it into 5 sets $D_{all} = \{D_1, D_2, D_3, D_4, D_5\}$ of size *N*/5 each where *N* is size of corpus.
2. For i = 1 to 5 Do,
3. Divide D_{all} into two subsets $D_{heldout} = D_i$. and $D_{train} = D_{all} - D_i$.
4. *M* = train the LM on D_{train}.
5. *Scores* = test *M* on $D_{heldout}$.
6. T_{min} = Find minimum score from *Scores*.
7. $S_i = T_{min}$.
8. End For
9. *T* = AVG(*S*). //average on 5-fold cross validation.
10. return *T* . //*T* is the final threshold selected

5. Experimental Results

This work implemented three different n-gram based statistical language model (LM) namely Witten-Bell (WB) LM, Backoff (BO) LM and proposed Witten-Bell Backoff (WBB) LM using the same setup and methodologies as explained section 4. In order to avoid model over fitting the training corpus was divided into two sets namely training set comprised of 80% of the data and held-out set

with rest of the 20% data. The trained LMs are tested on the held-out to find the best threshold to detect the valid and sentences as explained in section 4.4. To test the different LMs to detect grammatical validity of Bangla sentences, a set of 10000 correct sentences were collected distinct from the training data. Another 10000 ill-formed or invalid sentences were auto generated using insertion, deletion, transposition and substitution operations on the valid sentences. By inserting a word from a word list $W = \{w_1, w_2, ...w_m\}$ at $(n+1)$ positions, $m \times (n+1)$ sentences can be generated. Removing a particular word at a time from a sentence with n words, a total of n sentences can be generated each with $(n-1)$ words. By exchanging two consecutive words in a sentence, $(n-1)$ sentences can be generated allowing only one exchange at a time. Substituting each word once with its most possible l cohorts one can produce $(n \times l)$ sentences from a sentence with n words. Thus, using insertion, deletion, substitution & transposition operations, approximately $[m \times (n+1) + n + (n-1) + (n \times l)] \times r$ invalid sentences can be generated from a set of r valid sentences. This method will produce a huge number of sentences. The selected 10000 invalid sentences were randomly collected from these auto-generated sentences filtered with lower n-gram scores.

The comparative performance of the LM methods has been evaluated by Precision (PRC) and Recall (REC) which are calculated in terms of True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) [16]. Precision and Recall for positive or correct sentences can be defined as,

$$PRC_{pos} = \frac{TP}{TP + FP} \quad (12)$$

$$REC_{pos} = \frac{TP}{TP + FN} \quad (13)$$

Precision and Recall for negative or incorrect sentences can also be defined analogously as,

$$PRC_{neg} = \frac{TN}{TN + FN} \quad (14)$$

$$REC_{neg} = \frac{TN}{TN + FP} \quad (15)$$

Since FP and FN are quite dangerous for grammar verification systems, PRC and REC were selected for performance evaluation. Table 1 shows the comparative performances of the LM systems for all n-gram orders for both valid and invalid test sentences respectively.

As can be seen, the performance of each of the methods improves with the order of n-gram i.e. they perform well with higher order n-grams. This can be easily derived from the fact that each method performs their best with the 4-gram model.

Table 1: Comparative performance Analysis of Different LM Systems

Results attained with Valid Sentences						
LM Methods	**Precision(PRC_{pos})**			**Recall (REC_{pos})**		
	2-gram	***3-gram***	***4-gram***	***2-gram***	***3-gram***	***4-gram***
WB (existing)	59%	67%	78%	76%	78.5%	81%
BO	56%	68%	76%	76%	80%	81.5%
WBB	62%	71%	80 %	78%	82%	84%
Results attained with Invalid Sentences						
LM Methods	**Precision (PRC_{neg})**			**Recall (REC_{neg})**		
	2-gram	***3-gram***	***4-gram***	***2-gram***	***3-gram***	***4-gram***
WB (existing)	72%	80%	81%	32%	67%	76%
BO	70%	79%	81.5%	31%	66%	77%
WBB	74%	81%	83%	32%	68%	78.5%
Average Precision & Recall for All Methods						
LM Methods	**Precision (PRC_{avg})**			**Recall (REC_{avg})**		
	2-gram	***3-gram***	***4-gram***	***2-gram***	***3-gram***	***4-gram***
WB (existing)	66%	74%	80%	54%	73%	79%
BO	63%	74%	79%	54%	73%	79%
WBB	68%	76%	82%	55%	75%	81%

As can be noticed from the Table 1, the precision for the grammatical data are quite low compared to recall values for all models. Whereas, the recall values for the ungrammatical sentences are quite low compared to precision values for all LMs. This is due to the fact that there was a higher number of FPs found in the experiments. There may be two reasons that may have influenced the high FPs. Firstly, since due to unavailability of some standard real error corpus; the ungrammatical sentences were generated artificially. There may be some artificially generated error test sentences that may have still remained grammatically correct and hence detected as correct sentence increasing the FP rate. Table 2 shows some of the artificially generated invalid sentences that are actually valid grammatically which caused the classifier methods to misclassify them as FPs. Secondly, selecting lowest probability score among all correct sentences as threshold in 5-fold cross validation process has set a hard boundary for positive i.e. grammatical sentences reducing the number of FNs but adversely also increased the chances of FPs. The trade-off between precision and recall is depicted in Fig. 2 and Fig. 3 for both valid and invalid sentences respectively. Clearly as for all LMs the PRC increases with decreasing REC. The method that finds the best trade-off between them is the one most desirable for application. Following the results from Table 1 and Fig. 2 & 3, it's clear that the proposed WBB method outperforms both WB (existing) and BO methods in terms of precision-recall

trade-off achieving highest PRC of 80% and 83% for valid and invalid data sets respectively with its 4-gram model. It also attained the highest REC values for both correct and incorrect sentences with 84% and 78.5% REC values respectively. In terms of average PRC and REC among all test sentences the proposed method achieved the highest PRC of 82% and REC of 81% outperforming other methods. Table 3 & 4 presents some examples of correctly predicted valid and invalid sentences respectively.

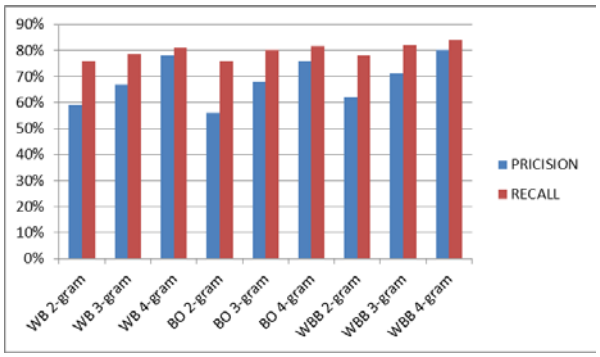

Fig 2. Precision and recall trade-off for valid sentences set for all LMs.

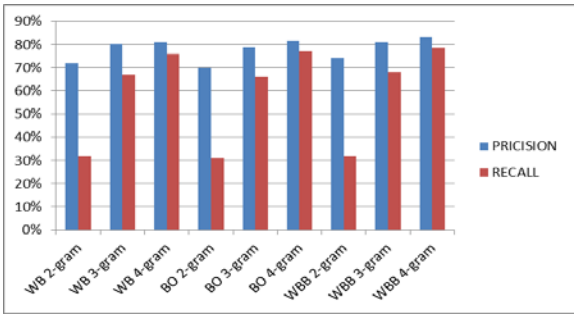

Fig 3. Precision and recall trade-off for invalid sentences set for all LMs.

Table 2: Examples of some valid sentences that are generated as invalid sentences & misclassified as FPs

তাঁদের ধারণা বিএনপি আমায়াতকে ছাড়লেই সরকারবিরোধী বৃহত্তর মোর্চা হতে পারে।
এতে এলাকায় আতঙ্ক ছড়িয়ে পড়লে পুলিশ গিয়ে পরিস্থিতি নিয়ন্ত্রণ আনে।
এই দিনেই আল্লাহ তাআলা এই পৃথিবীতে সর্বপ্রথম রহমতের বৃষ্টি বর্ষণ করবেন।
গত সোমবার এ ঘটনা ঘটে।
আমাদের পরীক্ষার পদ্ধতি ভালো নয়।

Table 3: Examples of some correctly predicted valid senetences by the proposed method

এ ঘটনায় কেউ গ্রেপ্তার হয়নি।
গত বছরের ডিসেম্বরে তিনি অবসরে যান।
পরিস্থিতি নিয়ন্ত্রণে আনতে পুলিশ শতাধিক ফাঁকা গুলি ছোড়ে।
প্রধানমন্ত্রী শেখ হাসিনার নির্দেশে এই সড়ক নির্মাণের পরিকল্পনা নেওয়া হয়েছে।
রাজধানীর বিভিন্ন সড়ক দিয়ে শোভাযাত্রা যাওয়ার সময় অন্যান্য সড়কে যানজট লেগে যায়।

Table 4: Examples of some correctly predicted invalid senetences by the proposed method

সেখানে কর্তব্যরত চিকিৎসক তাঁকে মৃত ঘোষণা ।
এই ভবিতে আমি কাজ তীবলতুড়ে করব।
এটা যে রাস্তা তা বোঝার উপায় না থাকে।
শারীরিক কারণে জামিনে থাকা কায়সার এ সময় ট্রাইব্যুনালে হাজির ছিলে।
তিনি ঢাকা বিশেষ জজ আদালত ৩ এর বিচারক হিসেবে কর্মরত ছিল।

6. Conclusions

In this work, a statistical Bangla sentence validity checking system has been developed which outperforms the existing systems for sentence grammaticality verification. As per our knowledge, this was first attempt to train and test the system on a large corpus of 10 million words for the purpose of grammar checking, which provided better clarity and generalization of performance measures. We expect that our attempt will encourage other researchers to work on Bangla grammar verification which needs further attention as development in this research area is not yet up to the mark. In future, we will try to combine some linguistic information into our statistical system for better performance.

References

[1] K. Jensen, G.E. Heidorn, S.D. Richardson, Natural Language Processing, the PL-NLP approach. 1993.
[2] D. Naber, A Rule-Based Style and Grammar Checker, Diploma Thesis, Computer Science. University of Bielefeld, 2003.
[3] C. D. Manning, P. Raghavan, H. Schütze, An Introduction to Information Retrieval. Cambridge University Press, 2009.
[4] TopTenReviews, “Online Grammar Check Reviews”, http://www.toptenreviews.com/services/education/best-online-grammar-checker/, Access Date: 28 October 2016.
[5] Y. Wu, “The Impact of Technology on Language Learning,” Future Information Technology, Lecture Notes in Electrical Engineering, vol. 309, pp. 727-731, 2014.
[6] J. Lane, “The 10 Most Spoken Languages in the World,” https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world, Access Date: 24 October 2016.

[7] Wikipedia, "Bengali language", https://en.wikipedia.org/wiki/Bengali_language, Access Date: 24 October 2016.
[8] D. Jurafsky, J.H. Martin, Speech and Language Processing An Introduction to Natural Language Processing: Computational Linguistics and Speech Recognition. Prentice Hall, Englewood Cliffs, New Jersey 07632 , September 28, 1999.
[9] C. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA: May 1999.
[10] K.M.A. Hasan, A. Mahmud, A. Mondal and A. Saha, "Recognizing Bangla Grammar Using Predictive Parser," International Journal of Computer Science & Information Technology (IJCSIT), vol. 3, pp. 61-73, December 2011.
[11] M.J. Alam, N. UzZaman, M. Khan, "N-gram based Statistical Grammar Checker for Bangla and English," Ninth International Conference on Computer and Information Technology (ICCIT), December 2006.
[12] N.H. Khan, M.F. Khan, M.M. Islam, M.H. Rahman and B. Sarker, "Verification of Bangla Sentence Structure using N-Gram," Global Journal of Computer Science and Technology: A Hardware & Computation, vol. 14, 2014.
[13] Wikipedia, "n-gram", https://en.wikipedia.org/wiki/N-gram, Access Date: 30 October 2016.
[14] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recogniser," IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3), 400–401, 1987.
[15] I.J. Good, "The population frequencies of species and the estimation of population parameters," Biometrika, 40 (3–4): 237–264, 1953.
[16] R.B. Yates, B.R. Neto, Modern Information Retrieval. New York, NY: ACM Press, Addison-Wesley, Seiten 75 ff., 1999.

Md. Riazur Rahman obtained his B.Sc. degree in Computer Science from Daffodil International University, Dhaka, Bangladesh. Now he is working as a Lecturer at the Department of Computer Science and Engineering in Daffodil International University. He is very keen on doing research work. He has a number of publications in international and national journals and conference proceedings. His research interest includes Natural Language Processing, Text Mining, Information Retrieval, Artificial Intelligence, Pattern Recognition, and Image Processing.

Md. Tarek Habib is continuing his Ph.D. degree at the Department of Computer Science and Engineering in Jahangirnagar University. He obtained his M.S. degree in Computer Science and Engineering (Major in Intelligent Systems Engineering) and B.Sc. degree in Computer Science from North South University in 2009 and BRAC University in 2006, respectively. Now he is an Assistant Professor at the Department of Computer Science and Engineering in Daffodil International University. He is much fond of research. He has had a number of publications in international and national journals and conference proceedings. His research interest is in Artificial Intelligence, especially Artificial Neural Networks, Pattern Recognition, Computer Vision and Natural Language Processing.

Md. Sadekur Rahman obtained his B.Sc. and M.Sc. degree in Applied Mathematics & Informatics from Peoples' Friendship University of Russia. Now he is working as an Senior Lecturer at the Department of Computer Science and Engineering in Daffodil International University. He has a number of publications in international and national journals and conference proceedings. His research interest includes Data Mining, Artificial Intelligence, Pattern Recognition, and Natural Language Processing.

Shaon Bhatta Shuvo obtained his M.Sc. degree in Computer Science from South Asian University, New Delhi, India in 2015. He obtained B.Sc. (Engineering) degree in Computer Science & Telecommunication Engineering from Noakhali Science & Technology University, Bangladesh. He is currently working as Lecturer at the Department of Computer Science &Engineering in Daffodil International University, Dhaka, Bangladesh. His research interest includesBig Data, Artificial Intelligence, especially Artificial Neural Networks, Natural Language Processing and Computer Vision.

Dr. Mohammad Shorif Uddin received his PhD in Information Science from Kyoto Institute of Technology, Japan, Master of Education in Technology Education from Shiga University, Japan, Bachelor of Science in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET) and also MBA from IBA, Jahangirnagar University. He currently serves as Professor and Chairman in the Department of Computer Science and Engineering, Jahangirnagar University, Dhaka. His research is motivated by applications in the fields of imaging informatics, computer vision and image velocimetry. He has published more than 70 papers in peer-reviewed international journals and conference proceedings and also delivered keynote speeches in some of international conferences in home and abroad. He holds two patents for his scientific inventions. He is the co-author of three books. He is a Fellow of Bangladesh Computer Society and also a senior member of IEEE and IACSIT.