Joint 2019 8th International Conference on Informatics, Electronics & Vision (ICIEV) &
3rd International Conference on Imaging, Vision & Pattern Recognition (IVPR)

Semantic Error Detection and Correction in Bangla Sentence

M. F. Mridha[*], Md. Abdul Hamid[‡], Md. Mashod Rana[€], Md. Eyaseen Arafat Khan[£], Md. Masud Ahmed[¥], Mohammad Tipu Sultan[#]
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
[*]firoz@uap-bd.edu, [‡]ahamid@uap-bd.edu, [€]mashod0rana@gmail.com, [£]eyaseenarafatkhan08@gmail.com,
[¥]mdmasudrana81uap@gmail.com, [#]tipu07u5@gmail.com

***Abstract*—Detection and correction of errors in Bengali text is essential. In general, Bengali text error can be classified into non-word error and semantic error (also known as context sensitive error). Till date, auto-correction for semantic error in Bengali sentence is challenging since there is no significant research works on this very topic. In this paper, we bring out the concept of Semantic Error detection and correction. We have developed a method that can detect and correct this kind of errors. Semantic error includes typographical error, grammatical errors, homophone errors, homonym error etc. Our goal to this study is to develop an approach to handle multiple semantic errors in a sentence. We have used our own built confused word list by edit distance and apply Naïve Bayes Classifier to detect and correct typographical and homophone error. For a candidate word from a sentence, we pick out a set of words which is a collection of confused words. We use all other neighbor words as features for each word from confusion set. Then we apply naïve theorem to calculate the probability and decide whether a target word is error or not. We have used 28,057 sentences to evaluate our model and we have achieved more than 90% accuracy. All data corpora used to evaluate the model are built by us. We strongly believe that the problem we have solved may shed light on the advancement of Bengali language processing significantly.**

Keywords-NLP; Naïve Bayes; Bangla; Semantic Error; Machine Learning.

I. INTRODUCTION

Writing is the most important way of communication for human. Writing represents the human language with sign and symbol and it works as a tool to make language readable. Writing is used as alternate of spoken language. Writing is not important only for communication but also for keeping records, publications, and storytelling etc. Writing helps us to passing history generation to generation, to maintaining culture etc.

Every language has a style to represent it in textual form. Bengali is the 7th most spoken language in the world with around 250 million peoples, which has its own sign and symbol in textual representation. At present it is important to process a language with computer system.

For official and non-official purpose, we do process our Bengali language in computerized system. Bengali language has critical grammatical rules and complex orthographical rules. That is why it is not soft to process Bengali language. When we are typing on the editor during chatting, mailing etc. usually we forget to maintain the rules of writing. That is why errors may occur frequently. And, importantly we cannot recognize them most of the time due to lack of knowledge. So, now it becomes a common expectation of auto correction in our text.

Error occurs in different level like word level, sentence level etc. When we find error in our spelling of a word it says word level error. But, when an error occurs semantically, we know it as sentence level error. That means, when a word is correct but not appropriate for this sentence in that position, we say it semantic error. It is also known as context sensitive error. Kukich [1] says about many types of error like real word error and non-word error. It is easy to detect non-word error. But detection of real word error is not so easy. In Bengali language, it becomes more difficult to detect when an error occurs in semantic level.

In Bengali language, context sensitive error can be categorized into homophone error, homonym error, typographical errors, grammatical errors etc. Here, we do focus our attention on typographical error. Typographical error also covers homophone error. Typographical error occurs during the typing. During typing, if we type extra character or miss to type character, it changes the word. But unfortunately, after change the word, if it remains correct then it destroys the context of the sentence. It may change the whole meaning of the sentence if error happens more than one. The following examples better explain this very important point.

Example-

সব ব্যাংকগুলো লুটপাট করে খাল(খালি) করেছে কারা

পুলিশের গু (গুলি) খেয়ে বকের (যুবকের)মৃত্যু

In the avobe sentences, some letter and sign are missing in underlined words, and they still are correct words, but the meaning of words are different. In the first sentence, like খাল means canal and খালি means empty, where the diffrence between them is one vowel sign. In the second sentence, the word গু for

978-1-7281-0788-2/19/1.00 ©2019 IEEE

faeces, গুলি for bullet where the difference is a vowel sign; the word বকের for the egret, যুবকের for youth, wherein wrongly used word is missing যু . And this type of error is known as deletion error. Though the words are correct they change the meaning of the sentence and disturb the semantic structure.

Correct sentence : সবাই ভাল কাজের জন্য নেকী পাবে

Incorrect sentence: সবাই বাল (ভাল) অকাজের (কাজের) জন্য নেকী পাবে

In the incorrect sentence in word বাল (Boy), ব take the place of the letter ভ (ভাল, Good) which known as replaced error. Homophone error can be occurred due to replaced error. In the word, অকাজের have extra letter অ which is known as insertion error. These two errors are destroying the sentence. Also, there has others error as like as homophone error where two words pronunciation are same but spellings are different also meaning. Example, ধান (paddy), দান (donation); For the following sentence the word ধান are correct, not দান. দান (ধান) চাষ কর।

When a set of words pronunciation and spellings are the same but meanings are different, then these are called homonym words and error caused for this type of words is known as homonym error. The following table contains the different types of error in Bengali sentence.

TABLE I. DIFFERENT TYPES OF ERROR

Sentences with different Error
আমার পাছায়(ছাপায়) চুল (ভূল) ছিল Replaced error: (পাছায়-In the back; ছাপায়-Print out), (চুল-hair; ভূল-mistake)
পরীক্ষায় কৃতকার্যকারীকে(অকৃতকার্যকারীকে) পুরষ্কৃত করা হয় না Deletion error: (অকৃতকার্যকারীকে-Failure who; কৃতকার্যকারীকে- The successor)
অসৎ (সৎ) লোক সম্মানের চাবিদার (দাবিদার) Insertion error: (অসৎ-dishonest; সৎ-honest) and Replaced error: (চাবিদার-key keeper; দাবিদার-Claimant)
আহিংসা পরম ধর্ম (religion) মানুষ ও পশুর ধর্ম (behavior) পৃথক Homonym error

In this research, we are going to detect the sentence level error. We address the issue when a word is correct for the sentence but the sentence has lost its original expression, and how this word can be replaced by another appropriate word. We also solve the problem when an error occurs more than one in a single sentence. In this topic, many research works have been done in other languages mostly in English. However, in the Bengali language, very few works are done and these are not enriched. And to the best of our knowledge we are the first who are going to handle multiple semantic errors in a sentence by the help of Naive Bayes Classification.

The rest of the paper is described as follows. Literature Review is presented in Section II. Section III contains the Proposed Method. Section IV will describe the whole methodologies in details. Handling Strategy of Multiple Error is presented in Section V. Section VI describes performance evaluation. The outcome of performance evaluation is described in Section VII. Section VIII concludes our work along with future research directions.

II. RELATED WORK

Many methods have been developed in NLP to solve the sentence-level error. The most popular methods are statistic based and rule-based approaches. In the rule-based approach, rules are made to solve the problem and rules are different for different languages. The statistical approach is quite more popular than rule-based approach due to its language independence.

Yves, Andrew and Golding [2] have given a method to real word error in a sentence by combining Trigram and Bayes theorem. The trigram is used for POS tagging and Bayes theorem is used for feature extraction.

M. Kim, S. Choi, H. Kwon [3] proposed a method which is the combination of Naive Bayes Classifier and Chi-Square methods to solve the context-sensitive error in Korean texts. They have tried to solve typographical error only. Islam et al. [4] [5] developed a trigram-based context-sensitive error by the help of the self-developed string similarity measure. Y. Bassil and Md. Alwani [6] proposed a method which is a blending of three algorithms and these are unigram, bigram and 5-gram to solve the context-sensitive error. Church and Gale [7] suggested the noisy channel for detection and correction for real word error which maintains the semantic characteristics of a sentence.

In the Bengali language processing, exact works are not done yet for semantic error. Similar and nearest works are done in the field of the spelling checker. Mentionable works are done for detection and correction of non-word error. However, as stated earlier, there are no significant research works done in the field of the context-sensitive or semantic level error. B. B. Chaudhuri has used an approximate string matching algorithm to detect non-word error [8]. Direct dictionary lookup method is used by N. UzZaman and M. Khan [9] for misspelt word error and by Abdullah and Rahman [10] to detect the typographical and cognitive phonetic error. P. Mandal and B. M. M. Hossain [11] proposed a method based on PAM clustering algorithm and this method also did not deal with the semantic error.

A few works are done at the semantic level. N. Hossain [12] have introduced with a model which used n-gram to check whether a word is a word or not in sentence level. K. M. Hasan, M. Hozaifa and S. Dutta [13] have developed a rule-based method which detects the grammatical semantic error in a simple sentence.

In this research, we are going to develop a method to detect and correct a semantic level error like typographical and homophone error. We are the very first to detect multiple semantic errors in a sentence with the help of Naïve Bayes Classifier.

III. PROPOSED METHOD

Processing Bangla language is not so easy. We are going to use the Naïve Bayes classification. We use a confusion set of words which will be made with an edit distance algorithm. Every

confuses word work as a single class. Then we will use conditional probability which will give us a score. The score calculated using conditional probability will help us to decide whether a word is appropriate or not and find the expected word. We also use Laplace Smoothing to get a better result.

IV. METHODOLOGIES

To build the proposed method we need to follow some step which will help us to achieve our goal. The steps are

- Collection of data
- Data preprocessing
- Extraction of the confused word list
- Applied Naïve Bayes theorem
- Declaration error and suggestion

A. Collection of data

To evaluate any framework, lots of data are required. In NLP data is an important fact to justify any method. More data will help us to prove how good a method works. That is why we have justified our method by considerably large data. We have collected data from the web, from the newspapers which are available online, from many blogs etc. We store them in a different file. The entire data corpus contains many types of data like political, fictional, sports, entertainments etc.

B. Data preprocessing

When we have collected data, these were not in the format which is needed for our method. So we need a preprocessing to get the expected format. We write python code to remove unnecessary sign and symbol. We also remove the emoji. Then we break them into sentences using Bengali punctuation rules and store them in a file. From these processed data we collect the unique words which we have used as our dictionary. We also collect words for our dictionary manually.

In our method, we are going to generate a set of confusion words for the target word. We preprocess the confused word by using our dictionary which is the collection of unique word. For each word, we try to bring the words which can be possible confused words. We apply the edit distance algorithm to create the set. We did not pick out confused word set for stop words. In our method occurrences of words in a sentence, occurrences of a word with its neighbor words etc. are going to be used for the purpose of calculating the probability. That is why we also preprocess the occurrences of a word for the corpus and create a corpus which is the collection of counts of a word with others words.

C. Extraction of the confused word list

In this step, we take all unique words from our dictionary and for every unique word we extract its confusing word set by the help of edit distance algorithm.

Edit distance algorithm is a way to find the minimum number of operations to transfer a string to another string; where operations are the insertion, deletion and replacement. It is also known as Levenshtein distance. We use the minimum distance 2. If any word takes less or equal to 2 operations to transform our target word we take this as a confuse word corresponding of a target word. We have created a collection of the confused words set as a file for every unique word in advance. Then, we extract the confused word set for the target word.

We take a sentence as input. We have assumed that there has no non-word error in our input sentence. From the input sentence, for every word we try to obtain a list of confused words by searching in our corpus which is the collection of list of confused words.

Let's our input sentence is consisting of n words.

$$\text{Input sentence, } IS=\{W_1,W_2,W_3,\ldots\ldots, W_{n-1},W_n\} \quad (1)$$

For i^{th} word W_i we will find a confused word set. If the number of confused words is m, then the set of confused words, SCW, is

$$\text{For } W_i \text{ word } SCW=\{cw_1, cw_2,cw_3,\ldots\ldots, cw_{m-1},cw_m\} \quad (2)$$

Where cw_j will be the j^{th} confused word from the SCW (means set of the confused word). Since our target word can generate a confused word list, it means that there is a chance of occurrences of semantic error due to deletion error or insertion error.

D. Applied Naïve Bayes theorem

In this step, we would find a list of confused words. If we have no confused words list, then we can declare the target word as error free. However, in the case of confused words list, we follow some procedure to decide the result. Here we are going to use Naïve Bayes classifier for deciding which confuse word is going to fit our sentence.

Naïve Bayes classifier is a model to classify in machine learning which is based on Bayes theorem. Bayes theorem simply follows the conditional probability as described in the following.

For a W_i from IS, Bayes theorem can be written in a modified way

$$P(W_i|W_1W_2\ldots W_{i-1}W_{i+1}\ldots W_n)$$

$$=\frac{P(W_1|W_i)P(W_2|W_i)\ldots P(W_{i-1}|W_i)P(W_{i+1}|W_i)\ldots P(W_n|W_i)P(W_i)}{P(W_1)P(W_2)\ldots P(W_{i-1})P(W_{i+1})\ldots P(W_n)} \quad (3)$$

Equation (3) can be written as

$$P(W_i|W_1W_2\ldots W_{i-1}W_{i+1}\ldots W_n) \propto P(W_i|W_1W_2\ldots W_{i-1}W_{i+1}\ldots W_n)P(W_i)$$

$$P(W_i|W_1W_2\ldots W_{i-1}W_{i+1}\ldots W_n) \propto P(W_i)\prod_{k=1}^{k=i-1}P(W_k)\prod_{k=i+1}^{k=n}P(W_k) \quad (4)$$

In (4), W_i will be replaced by every confuse word cw_j from the SCW because W_i is our target word. Here, W_i is our target word and others are used as feature words.

For feature word, it will not be good work to take all words as feature words from the sentence. When the distance is increased from the target word the semantic relation also

becomes weak with the neighbor word. So, to avoid weak semantic relation we set the neighbor distance 5 from the target word. We extract the target word features from the left side of the target word between distance 5 and from the right side of the target word.

Now for every cw_j from SCW the features set will be

$$FS=\{fw_1, fw_2, \ldots, fw_z\};\ 1<=z<=8$$

Now we will take fw_l which is l^{th} word in our features set FS and try to count occurrences of the fw_l word in our corpus with cw_j word. We will also count how many sentences contain the word cw_j from the total number of sentences in our corpus. And these counts will help to calculate the probability.

What happens if we don't find any occurrences of any features words with the confuse word cw_j. The probability will zero and which is not good for us. Cause we calculate probability from our corpus. It is sure that our corpus did not contain all possible sentence. That means there have chances that feature occurrences could be possible with word cw_j which is not in our corpus. To solve this problem, we use Laplace Smoothing to avoid this type of error.

Laplace Smoothing is a way to smooth classified data and also known as additive smoothing. And the Laplace Smoothing equation is given by

$$P(fw_l)=\frac{counts(fw_l)+\alpha}{counts(cw_j)+\alpha.counts(unique\ words)} \quad (5)$$

Where $counts(fw_l)$ is the occurrences of fw with cw_j; $counts(cw_j)$ is the total number of words in all sentences in corpus which contain the cw_j words; $counts(unique\ words)$ is the total number of unique words in the corpus; And α is always 1. $\alpha=0$ means no smoothing.

E. Declaration error and suggestions

After applying Naïve theorem and Laplace smoothing, we get the probability for all confused words from the SCW. And we can declare the target as the error or not on the basis of calculated probability. The word with higher probability will be the most appropriate word on that place and with less probability will be the less appropriate for that place. If our target word is not the highest probability word it will be declared as the error.

If we assume every cw_j word from SCW as a class, then we can represent Naïve Bayes classifier as

$$argmax_{SCW} P(SCW|FS)$$

$$=argmax_{SCW} P(FS|SCW)P(SCW)$$

$$=argmax_{SCW} \prod_{fw\ \epsilon\ FS} P(fw|SCW)P(SCW) \quad (6)$$

After the declaration of the error, the extracted confused word set will be provided as suggestion list. The SCW will be sorted by their probability. The word with maximum probability will be the top on the suggestion the list. Figure 1 presents the whole process. Following Fig 1 will represent the whole process.

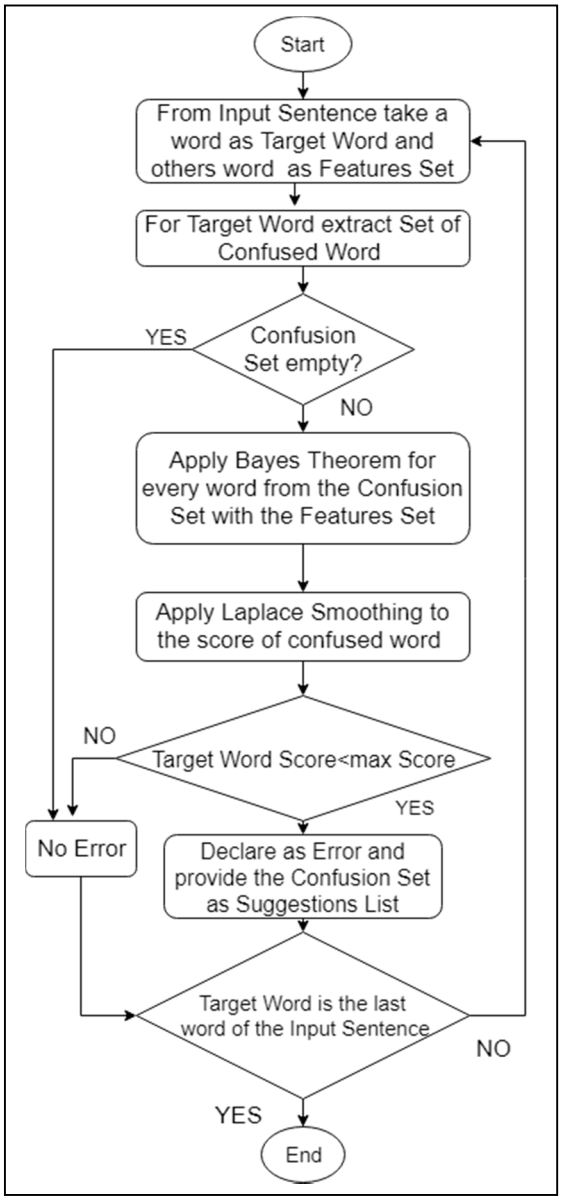

Figure 1. Flow Chart of Proposed Method

V. EVALUATION

To evaluate our proposed method, we build our own corpus. We build corpus by collecting data from the web. We have built 4 corpora to test our model. Data are formatted as line by following the rules of punctuations of Bengali Language. Then we take every sentence from the corpus as input and remove the stop words from the sentence. Then we evaluate our sentence through our model by Naïve Bayes Classifiers.

First, we have trained our model by using our collected data as a training set. Then we have injected error on our testing data corpus on the purpose of evaluation. Errors are injected to corpus randomly by using a written program. Because in practice it is quite difficult to collect the semantic error. Following Tables II-V are described with some incorrect sentence. The tables contain the suggestion words with their score. Since we decide to take

edit distance 2 to generate confusion set. That's why there has so many suggestions word and we take the number of words to show that our expected word is in the suggestion list.

The following table shows the evaluation for the sentence: পুলিশের **গু** (গুলি) খেয়ে **বকের** (যুবকের)মৃত্যু

TABLE II. FOR TARGET WORD গু(DELETION ERROR)

Suggested word	Score
গুলি	1.7627185390517586e-20
গাছ	3.409311006362797e-21
গতি	1.5746065463027899e-21
গড়া	1.9268956217462368e-22
গুল	3.517054829170595e-23
গুন	3.508881970897087e-23
গু	1.7681087832799074e-23

TABLE III. FOR TARGET WORD বকের (DELETION ERROR)

Suggested word	Score
যুবকের	1.5688367185562306e-22
বাকের	1.43723710964709e-22
বুকের	7.026839026529137e-23
পদকের	7.025022836560187e-23
বরের	3.518419284740286e-23
বকের	1.7681087832799074e-23

Following table are show the evaluation for the sentence:

সবাই বাল (ভাল) অকাজের (কাজের) জন্য নেকী পাবে

TABLE IV. FOR TARGET WORD বাল (REPLACED ERROR)

Suggested word	Score
ফেল	2.28311850932232e-27
ফজল	2.2816419573851163e-27
খাই	2.2805352953946707e-27
ভাল	2.2797978785685965e-27
বাল	5.722590488655557e-28

TABLE V. FOR TARGET WORD অকাজের (ADDITION ERROR)

Suggested word	Score
কাজে	1.0793053263947006e-24
কাজের	1.9437926723830688e-25
কাজেই	1.6414089955483 99e-25
কালের	1.2209853377939654e-26
অকাজের	5.722590488655557e-28

In some case, our expected word too below to our list. It happens for lack of data. More frequently happens words go top of the list. This a limitation of our current method. The overcome technique is mentioned in the part of the conclusion. Another reason is when there has more than one error in a sentence, during the processing of one error another error is taken as context error which has an effect in processing error. But when user will select the correct word for the sentence than another error will get correct context word which will minimize method error.

We have used four corpora which contain a total of 28057 sentences to test our model. We have gained 90% accuracy on average in our trained corpus. Table II shows the entire performance outcome on testing data.

TABLE VI. PERFORMANCE ON TRAINING DATA

Name of the Dataset	No of Sentence	No of Error Word	No of Detected word as Error	Accuracy
Corpus 1	7115	6160	5609	91.05%
Corpus 2	8156	7124	6356	89.21%
Corpus 3	8038	6702	6089	90.85%
Corpus 4	4748	4001	3603	90.05%
Total	28057	23987	21657	90.28%

The accuracy can be gone down if data are not found in the occurrences corpus. Since there have lots of data, it may happen that our collected corpus does not contain the target word. Also, it can happen that we did not find the confusion set in our method where in practice there exists confusion set for the corresponding target word. That is why we use Laplace Smoothing to handle the absence of the word in our occurrences corpus. Since there has no available rich open source corpus in Bangla we can't evaluate our method against other corpora. Also as far as our knowledge, in Bangla existing systems like Avro and others, they do not handle semantic error seriously, they are specialized at spell checking. That's why we are not providing a comparison with other existing systems.

VI. CONCLUSION

In this study, we have put effort to solve the typographical error and homophone error, which destroy the context of a sentence in the Bengali language. We have solved not only single error but also more than one errors in a sentence. Though we have achieved a good level of accuracy, there exists challenges and scope to improve. Because when multiple errors arise, it increases the time and space complexity which is not a good characteristic of a model. In future, we will try to develop a stop words corpus and try to tf-idf (term frequency and inverse document frequency) which will provide the better influence of context word. Also, a method will be developed in future to decrease the time and space complexity.

ACKNOWLEDGEMENT

This research is supported by The Institute for Energy, Environment, Research and Development (IEERD), University of Asia Pacific (UAP).

REFERENCES

[1] K. Kukich, "Techniques for automatically correcting words in text," ACM Computing Surveys, 24 (4), page 377 - 439, 1992.

[2] Golding and Andrew, "A Bayesian hybrid method for contextsensitive spelling correction," arXiv preprint cmp-lg/9606001, pp 1-15 (1996).

[3] M. Kim, S. K. Choi and H. C. Kwon, "Context Sensitive Spelling Error Correction Using Inter Word Semantic Relation Analysis," 2014 International Conference on Information Science & Applications (ICISA), page 1-4, 2014.

[4] A. J. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," ACM Transactions on Knowledge Discovery from Data, vol. 2, no.2, pp. 1–25, 2008.

[5] A. Islam and D. Inkpen, "Real-word spelling correction using Google web 1T 3-grams," Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, vol. 3, pp. 1241–1249, 2009.

[6] Y. Bassill and M. Alwani1, "Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information" Computer and Information Science, Vol. 5, No. 3, May 2012

[7] K. W. Church and W. A. Gale, "A spelling correction program based on a noisy channel model," COLING '90 Proceedings of the 13th conference on Computational linguistics – Volume 2, Pages 205-210,1990.

[8] B. B. Chaudhuri, "Reversed word dictionary and phonetically similar word grouping based spell-checker to Bangla text", Proc. LESAL Workshop, Mumbai, 2001.

[9] N. UzZaman and M. Khan, "A comprehensive bangla spelling checker," In the Proceeding of the International Conference on Computer Processing on Bengali (ICCPB), Dhaka, Bangladesh, 2006.

[10] A. Abdullah and A. Rahman, "A generic spell checker engine for south asian languages," Conference on Software Engineering and Applications (SEA 2003), 2003, pp. 3–5.

[11] P. Mandal and B. M. M. Hossain, "Clustering-based Bangla Spell Checker," 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Page 1 – 6, 2017

[12] N. H. Khan, G. C. Saha, B. Sarker and M. H. Rahman, "Checking the correctness of Bangla words using n-gram," International Journal of Computer Application, vol. 89, no. 11, 2014.

[13] K. M. A. Hasan, M. Hozaifa and S. Dutta, " Detection of Semantic Errors from Simple Bangla Sentences," 2014 17th International Conference on Computer and Information Technology (ICCIT), pages 296 – 299, 201.