2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 February, 2019

An Approach for Detection and Correction of Missing Word in Bengali Sentence

M. F. Mridha
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
firoz@uap-bd.edu

Md. Mashod Rana
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
mashod0rana@gmail.com

Md. Abdul Hamid, *Member, IEEE*
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
ahamid@uap-bd.edu

Md. Eyaseen Arafat Khan
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
eyaseenarafatkhan08@gmail.com

Md. Masud Ahmed
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
mdmasudrana81uap@gmail.com

Mohammad Tipu Sultan
Department of Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
tipu07u5@gmail.com

***Abstract*—Auto-correction for missing word in a sentence is not so easy. Also, it is found more challenging for the Bengali language. Our rigorous study reveals the fact that no significant research works have been done for the Bengali Language on this very topic. In this paper, we proposed a method that can detect the missing word and provide a suggestion list correspond to missed word with 82.82% accuracy. We have used n-gram model to find whether a word is missing between two words from a sentence or not. Then, we have used probability scoring to rank the suggestion list after finding the probable words for the missed word. We have used a corpus for making the decision which is the collection of bigram and another corpus is used for preferable word for missed word which is a collection of the trigram. Finally, we have used another six corpora to evaluate our proposed method. All corpora are created by us using the data collected from the web.**

***Keywords*—NLP, N-gram, Missing Word Error, Bengali Language.**

I. Introduction

We communicate with each other through languages. Bengali is the primary language in Bangladesh and the second most spoken language in India. Bengali is one of the most widely spoken languages with around 250 million people throughout the world. To communicate and keep records in the purpose of official and nonofficial, we use textual representation. In the computerized system, it is not so easy to process the Bengali Language because of its complex orthographical rules and critical grammatical rules that are quite so hard to follow. That is why it becomes a common expectation for auto-correction in our text which is known as spelling correction.

The duty of a spell checker is to detect errors and also to provide the suggestions. The absence of a word in our known dictionary is a spelling error. Kukich [1] has said about two types of error: non-word error and real word error. However, in this paper, we are not going to work with these types of error. We are focusing on how to detect the word which is unintentionally missed during typing in a sentence. In this paper, we call this missing word error.

Missing word error is a sentence level error. This excessively occurs when we are typing a large article. It also occurs during conversion of OCR. The example shown below helps us to understand the error.

Example:

“এই জেলায় তার চা বাগান থেকে সবচেয়ে ভাল চা তৈরি হয় এবং সেখানে কখনও শ্রমিক বিক্ষোভ হয়নি। এই না হওয়ার পেছনে দুটো কারণ আছে। মালিক বলেন, মালিক হিসেবে তিনি খুবই পেশাদার। তিনি ভাল বেতন ও সুযোগ সুবিধা দিয়ে থাকেন। নিন্দুকের বক্তব্য হল, শ্রমিকদের মধ্যে যারা নেতা হতে চায় তাদের তিনি কিনে ফেলেন বা সরিয়ে দেন।”

This is a correct paragraph in Bengali described above. Now, if anyone types this paragraph is shown below:

“এই জেলায় তার চা বাগান থেকে সবচেয়ে চা তৈরি হয় এবং সেখানে কখনও শ্রমিক হয়নি। এই না হওয়ার পেছনে দুটো কারণ আছে। মালিক বলেন, মালিক হিসেবে তিনি খুবই পেশাদার। তিনি ভাল বেতন ও সুযোগ সুবিধা থাকেন। নিন্দুকের বক্তব্য হল, শ্রমিকদের মধ্যে যারা নেতা হতে চায় তাদের তিনি ফেলেন বা সরিয়ে দেন।”

If we match these two paragraphs, we can easily realize that some words are missing from the second one. And the words are marked with red colour in the following paragraph:

“এই জেলায় তার চা বাগান থেকে সবচেয়ে <u>ভাল</u> চা তৈরি হয় এবং সেখানে কখনও শ্রমিক <u>বিক্ষোভ</u> হয়নি। এই না হওয়ার পেছনে দুটো কারণ আছে। মালিক বলেন, মালিক হিসেবে তিনি খুবই পেশাদার। তিনি ভাল বেতন ও সুযোগ সুবিধা <u>দিয়ে</u> থাকেন। নিন্দুকের বক্তব্য হল, শ্রমিকদের মধ্যে যারা নেতা হতে চায় তাদের তিনি <u>কিনে</u> ফেলেন বা সরিয়ে দেন।”

Usually, during typing, unintentionally we give up some words, usually more on a large article and also it happens when processing data by automated software. As stated earlier, we are going to say this as missed word error and this not an unfamiliar issue to us. This is a common occurrence in our daily working who works in the text editor and similar services. It also happens during a chat, writing e-mail etc. So this is not an issue that may be ignored.

This fact drives us to propose a method to solve this missed word error problem. To the best of our knowledge, no research work has been done on this topic in the Bengali Language. The rest of the paper is organized as follows. In Section II similar work is discussed. In Section III proposed framework of missed word error is described. Section IV describes our methodologies. Section V and VI present our analysis and results, respectively. Finally, Section VII concludes our works.

II. Similar Work

During the study, we have been trying to extract the method that has already used to solve this type of problem. But we did not get any satisfactory result. The resources are not rich to solve exactly this type of problem. We have found a work which is done by Ahmed BEN SALAH [2]. They tried to detect the missing component which would be text or graphics in OCR output. They took the empty area and background since the error could occur only there. They learnt each element detected by OCR and found the nearest element for the targeted area. Our work is somewhat similar to this work. However, we are going to detect the missing word from a sentence which occurs not only after conversion by OCR but also when a man does this type of error unintentionally during the typing in the text editor.

In Bengali, this type of work is not done yet as far our study reveals, but numerous works have been done on the purpose of spelling check. Bidyut Baran Chaudhuri [3] developed a model which can deal with non-word error in Bengali sentence. UzZaman and Khan [4] proposed a Double Metaphone encoding system which was used for Bengali. This encoding can be used to improve spell check in Bengali. It was only used for the misspelled word. Nur Hossain Khan [5] used characters of n-gram to check the correctness of Bengali words and there was no specification about missing words identification. Prianka Mandal [6] developed a clustering based Bengali spell checker which is also for non-word error, where Partitioning Around Medoids algorithm is used. We are going to detect the missing words by using n-gram model with the help of probability. N-gram is used in [7] and in [8], to detect real word error in English language and for other NLP purpose. Therefore, it is the very first work on the Bengali language to detect the missing word in a sentence using n-gram.

III. Proposed Framework

To build the method we need: (i) First, we need to decide whether a word is missing or not, and (ii) If missed, we should provide a suggestion list against the missed word.

From a sentence first we will generate bigrams. If a bigram exists in our corpus, then it is alright, otherwise, it declares as an error. To provide suggestion list, we search in our trigram corpus where bigram first and the last word be trigram first and last word respectively and middle word be our suggestion word which is the missing word.

IV. Methodologies

Our methodologies consist of the following steps: (i) Data pre-processing, (ii) Generate n-gram, (iii) Detection of the missed word, and (iv) Find the suggestion list (if an error exists). We describe each one in the following.

A. Data Preprocessing

We have collected the data from the various online newspapers, blog etc. We have stored data in six different files known as the corpus. From these corpora, by removing the unnecessary symbol we generate bigram and trigram. Then we build two corpora: one is the collection of bigrams with their frequency (count) and another one is the collection of trigrams with their frequency.

B. Generate N-gram

N-gram is the n numbers of contiguous elements from a sequence of elements. Where elements would be characters, words, speech, text etc. In natural language, n-gram was first introduced by Shanon [9]. The importance of n-gram is that it is language independent [10]. When n=1 it is known as the unigram, bigram for n=2 and trigram for n=3.

If a sequence is a collection of n elements, then the bigram and the trigram of the ith word will be

$$\textit{Sequence of elements} = \{\boldsymbol{e_1, e_2, \ldots\ldots, e_{n-1}, e_n}\}$$

Where $\boldsymbol{e}$ for elements.

$$\textit{Bigram: } \boldsymbol{e_i e_{i+1}}$$

$$\textit{Trigram: } \boldsymbol{e_i e_{i+1} e_{i+2}}$$

For our model, we take a sentence which consists of n number of words. And, we also assume that there have no non-word errors and real word errors in our sentences.

$$\textit{Sentence} = \{\boldsymbol{w_1+w_2+\ldots\ldots+w_{n-1}+w_n}\}$$

From this sentence, we will generate bigram. For ith word Bigram = W_iW_{i+1}

C. Detection of missing word

For every bigram that we generate from the sentence, we will search in our bigram corpus. If we find the bigram in the corpus, then there is no error. But if it fails to match in corpus then we say that there has been an error in the sentence. This means that we expect a word between these two words which is missed unintentionally. Since there is a missed word between these two words we expect the suggestions which will tell us which word could best fit in the place of the missed word.

D. Finding the suggestion list

When it is decided that a word is missing between the bigram, we need the suggestion list. To extract the suggestion list we take the bigram and from our trigram corpus, we extract the word list with frequency. In this step, we try to find these trigrams where trigram first word is same as our bigram first word and trigram the last word is same as our bigram the last word.

If trigram is $TW_1TW_2TW_3$ (three words)

Then,

TW_1 = bigram first word.

TW_2 = expected missing word.

TW_3 = bigram last word.

For each trigram that we extract from the trigram corpus, we will calculate the probability using their frequency. The extracted trigram list will be sorted and higher probability trigram will be in the top of the list and only 2nd word of the trigram will be shown on the list. We will use tg as trigram in our equation.

$$\textit{List of extracted trigram } (tg) = \{tg_1, tg_2, \ldots\ldots, tg_z\}$$

For each trigram,

$$p(tg_i) = \frac{frq(tg_i)}{\sum_{j=1}^{z} frq(tg^j)} \quad (1)$$

Where, frq(tg_i) is the frequency of ith trigram from the list, and the denominator of the equation is the sum of every trigram frequency contains from the list. In our original code, we do not need to save trigram in the trigram list which also known as suggestion list. Because we just need the missing word that's why we will store the middle word of the trigram and the trigram frequency.

The following steps are summarizing the procedure: Bigram=B_1B_2; where B1=first word B2=second word Trigram=$T_1T_2T_3$; where T_1=1st word T_2=2nd word T_3=3rd word.

```
Generate a set of bigram SB from a sentence
For 1 to n in SB where n is the number of bigram SB
    If ith $B_1B_2$ is present in bigram corpus
        No missing word error
    Else
        In ith $B_1B_2$ a word is missing
        S=suggestion list
        Search $T_1T_2T_3$ in the trigram corpus
            If $T_1=B_1$ and $T_3=B_2$
                Insert the $T_2$ in S with frequency
        For each word in S calculate score using (1)
Sort the suggestion list S in decreasing order by score and show them.
```

V. Analysis

For the evaluation purpose, we have created six corpora where data are collected from the web. To evaluate our method, we injected the errors in our corpora. That means we write a program to delete a word from the sentence in our all corpora. After making error containing corpus, we take the corpus as input and we identify the sentences. From a sentence, we generate bigram and using bigram corpus to identify whether it is correct or not. To provide suggestion list to the corresponding error we used trigram corpus.

Here we have presented a description to explain our proposed model with some sentences. We will show how it performs with the example given below:

Right: তিনি বিদঘুটে নামটা কিছুতেই মনে করতে পারেন না।

Wrong: তিনি বিদঘুটে নামটা কিছুতেই করতে পারেন না।

Where the word "মনে" is missing in the wrong sentence shown above. Our proposed method, the bigram "কিছুতেই করতে" did not find in the bigram corpus and declares as the error. Then it searches for trigram in trigram corpus where it extracts every matched trigram with possible word that can be inserted between bigram to find missing word and calculate the probability to generate suggestion list Table I shows the suggestion list with the score.

TABLE I. Suggestion list for bigram "কিছুতেই করতে"

Suggested word	Score
মনে	0.2857
বিশ্বাস	0.1428
বশ	0.1428
প্রকাশ	0.1428
জুত	0.1428
কল্পনাও	0.1428

The table shown above provides the correct answer through the list.

Right: গত কয়েক সপ্তাহ ধরে দেশের প্রধানমন্ত্রি টেলিভিশন, রেডিও এবং সংবাদপত্রে বার বার বিবৃতি দিয়েছেন, কিছু ঘটবেনা, সবাই শান্ত থাকুন।

Wrong: গত কয়েক ধরে দেশের প্রধানমন্ত্রি টেলিভিশন, রেডিও এবং সংবাদপত্রে বার বার বিবৃতি দিয়েছেন, কিছু ঘটবেনা, সবাই থাকুন।

TABLE II. Suggestion list for bigram "কয়েক ধরে"

Suggested Words	Score
বছর	0.3076
দিন	0.2307
সপ্তাহ	0.1538
মাস	0.1538
শতক	0.0769
দশক	0.0769

The word "সপ্তাহ" is correct in this list.

TABLE III. Suggestion list for bigram "সবাই থাকুন"

Suggested Words	Score
ভাল	0.333
সুখে	0.333
শান্ত	0.333

The word "শান্ত" is the correct word in this list. But it's rank falls below in the list because the frequency is the same for all consisting trigram than any other word which is in the list.

VI. Result

Our bigram corpus contains 4,00,000 bigrams and trigram corpus contains 4,43,000 trigrams. We have used the calculation of probabilities for ranking the words and rank is used to provide the suggestion.

After applying the method to our all corpora we get the average 82.82% accuracy. On an average, every corpus contains 15,000 sentences. The detailed about the experiment for different corpus are listed below in Table IV.

TABLE IV. Performance Evaluation

Dataset Name	No of error	No of error as detected and able to provide suggestion	Accuracy
Corpus 01	17730	14387	81.14%
Corpus 02	15017	12419	82.70%

Corpus 03	13523	11448	84.66%
Corpus 04	16578	13653	82.36%
Corpus 05	7853	6771	86.22%
Corpus 06	27059	22282	82.35%
Total	97760	80960	82.82%

The lowest accuracy is around 81% and the maximum accuracy is around 86% what we gained. There were some reasons that we cannot gain higher accuracy. It could be that a word is missing from the start of the sentence or end of the sentence. However, in this paper, we did not consider this type of cases. Since, usually during typing or chat, we do not miss our starting word, we ignore this type of cases. There are some types of cases that after missing a word from a sentence it is still a correct sentence. In that case, our method fails to detect the missed word error. We did not count that types of detection in our accuracy calculation. However, the method detected the error but failed to provide suggestion because of lack of data on the corpus. Therefore, at present, we are rigorously working on developing various corpora and incorporating deep learning to mitigate these limitations and improve the accuracy of missed word error in the Bengali language.

VII. Conclusion

In this paper, we have proposed an approach for missed word error detection and correction in the Bengali language. We bring out the concept of missed word error that is essentially different than non-word error and real word error in the context of Bengali language. Therefore, we have focused on how to detect the word which is unintentionally missed during typing in a sentence. In this research paper, we have developed a solution to this problem and achieved more than 82% accuracy. We strongly believe that this particular research area opens up the new possibilities for greater improvement of Bengali language. We are working on this topic to incorporate deep learning for accuracy enhancement. In future works, we plan to explore issues of time complexity, statistical analysis for detection accuracy, different corpus for training and testing.

References

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

[2] A. B. Salah, N. Ragot, T. Paquet and T. Paquet, “Adaptive detection of missed text areas in OCR outputs: application to the automatic assessment of OCR quality in mass digitization projects,” SPIE. Document Recognition and Retrieval XX, Feb 2013, SAN FRANCISCO, United States. 8658, pp.110-122, 2013.

[3] B. B. Chaudhuri, “Reversed word dictionary and phonetically similar word grouping based spell-checker to bangla text,” LESAL Workshop, Mumbai, 2001.

[4] N. UzZaman and M. Khan, “A double metaphone encoding for bangla and its application in spelling checker,” International Conference on Natural Language Processing and Knowledge Engineering. IEEE, pp. 705–710, 2005.

[5] 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.

[6] Prianka Mandal and B M Mainul Hossain, “Clustering-based Bangla spell checker,” 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1-6, April 2017.

[7] S. Sharmaa and S. Gupta, “A Correction Model for Real-word Errors,” Procedia Computer Science, vol. 70, pp. 99-106, 2015.

[8] P. Samanta and B. B. Chaudhuri, “A simple real-word error detection and correction using local word bigram and trigram,” Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013), pp. 211-220, October 2013.

[9] C. E. Shannon, “Prediction and entropy of printed English,” Bell system technical journal, vol. 30, no. 1, pp. 50–64, 1951.

[10] F. Ahmed, E. W. D. Luca and A. Nürnberger, “Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness,” Polibits, no. 40, pp. 39–48.