Spell corrector for Bangla language using Norvig's algorithm and Jaro-Winkler distance

Istiak Ahamed[1], Maliha Jahan[2], Zarin Tasnim[3], Tajbia Karim[4], S. M. Salim Reza[5], Dilshad Ara Hossain[6]
[1,2,3,4,5]Department of Information and Communication Technology, Bangladesh University of Professionals (BUP), Dhaka, Bangladesh
[6]Department of ICT, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia

Article Info

Article history:

Received Mar 23, 2020
Revised Apr 24, 2021
Accepted Jun 22, 2021

Keywords:

Edit distance
Norvig's spell corrector
String similarity

ABSTRACT

In the online world, especially in the social media platform most of us write without much regard to correct spelling and grammar. The spelling mistakes are much larger in proportion when it comes to Bangla language. In our paper, we presented a method for error detection and correction in Bangla words' spellings. Our system could detect a misspelled Bangla word and provide two following services-suggesting correct spellings for the word and correcting the word. We had used Norvig's algorithm for the purpose but instead of using probabilities of the words to prepare the suggestions and corrections, we had used Jaro-Winkler distance. The previous works done in this field for Bangla language are either very slow or offers less accuracy. Our system successfully achieved a 97% accuracy when evaluated with 1000 Bangla words.

Corresponding Author:

S. M. Salim Reza
Department of Information and Communication Technology
Bangladesh University of Professionals
Mirpur Cantonment, Dhaka- 1216, Bangladesh
Email: salim.reza@bup.edu.bd

1. INTRODUCTION

Misspelling is a common phenomenon, especially when it comes to the internet. Bad spelling makes a person appear less intelligent and less credible than they actually are. Spelling mistakes not only put a dent on someone's professional reputation, but also cost a fortune in sales and business. Spelling errors in medical packaging can be lethal and can end up costing someone's life. So checking for spelling mistakes and correcting them is a much needed service in each and every language.

There are many works on spell checking and giving correction or suggestions in other languages whereas a very few works on Bangla or Bengali language though Bangla is spoken by 230 million people as native speakers and by 37 million people as their second language. In this research paper, a process is proposed for the development of an effectual Bangla spell corrector using maximum two edit distance from the incorrect word and in addition we use distance algorithm instead of occurrence probability in a document which is used in Norvig's spell correction algorithm [1]. We collected the Bangla dictionary and letters (vowel and consonant and also vowel mark and consonant conjunct) from various sources. Our system first matches a word with the existing dictionary and in case of mismatch, our system gives a probable list of correct words and most probable word based on string similarity. This work also shows the performance and evaluation of our proposed method.

The arrangement of the rest of the paper is described here. Explanation of some technical terms and methods are given in section 2. Next comes section 3 which contains mentions of some of the related works in the field of spelling detection and correction. The details of our proposed approach is explained in the section 4, followed by section 5 where we explain the results achieved by our system and some discussions relating to our work is provided there. Finally, the paper ends with conclusion in section 6 and references.

In this paper, we address the issue of lack of proper spell checkers and correctors for Bangla language and suggest a possible solution to the problem. Although our work is not the first to address this issue, it certainly introduces a method that overcomes hurdles that the previous works failed to overcome. The previous works on this topic either offer less accuracy than ours or perform poorly when it comes to multiple error spelling mistakes. We propose a spelling corrector that can handle single and multiple errors and offers high accuracy of 97%.

2. BASELINE RESEARCH

In this section we will discuss about the terms that we used in this report. We will give an overview of Bangla language, type of errors in Bangla language, Norvig's spell corrector, string similarity.

2.1. Bangla language

The alphabets of Bangla language consists of 49 letters, where 11 being vowels and 39 being consonants. There is no uppercase or lowercase process in Bangla alphabet but there are some complex systems in Bangla language such as, Phonetically similar characters, consonant conjunct, *Phala*, *Matra*, vowel mark, modified symbol and many more [2], [3].

2.2. Types of errors

Kukich [4] classified misspelled word of 2 types, namely, real word mistake, and non-word mistake. Real-word mistake occurs when a correct spelling is used but the word is wrong based on the context. The latter, non-word mistake, is that type of mistake where the used word is neither a dictionary word nor a noun [5]. Non-word error is further divided into two classes-cognitive mistake and typographical mistake. Typographical mistake is simple errors like mistakenly adding or deleting or inserting or transposing characters. Table 1 shows the percentage of different types of typographical errors. Cognitive mistake is where the spelling is forgotten and typed in a similar phonetic manner [6].

Table 1. Various types of spelling mistakes in Bangla with their percentage [7]

Type of mistake	Percentage
Transposition mistake	5.27
Substitution mistake	66.32
Insertion mistake	6.53
Deletion mistake	21.88

2.3. Norvig’s spelling corrector

Norvig proposes an algorithm for spell correction [1] which determines the correctly spelled word out of all possible suggestion with the maximum probability of occurring in a data set.

$$argmax_{c\in candidates}\, P(c)P(w|c)$$

This expression has expresses Norvig’s algorithm and has four main parts. ‘*argmax*’ is the selection mechanism, ‘*P(c)*’ expresses the language model, ‘*c ε candidates*’ denotes the candidate model, and ‘*P(w|c)*’ denotes the error model. The candidate model makes some small edits to a word by adding a letter, exchanging two adjoining letters, taking out one letterand putting a different letter in place of a letter. For n^{th} length word, there are in total *54n+25* possibilities, which consists of *n-1* transpositions, *26(n+1)* insertions, *n* deletions, and *26n* alterations but with some duplication and only few are dictionary words. If the edit distance is 2 the suggestion list will be bigger and again few will be dictionary words. In terms of picking one single word as correct word from the suggestion list, the model in question uses probabilities of occurrence of those words belonging to the suggestion list. The probabilities of occurrences are derived from a test document to rank the candidates. The word from the recommended list having the highest probability becomes the correction for the misspelled word.

2.4. String similarity algorithm

According to the function and types of operation, string similarity algorithms can be categorized into few domains:

a. Edit distance based; edit distance based string similarity algorithm takes two words mostly of same length and compare with each other for the unmatched characters. Hamming distance [8], Levenshtein distance, Trigram comparison [9], Jaro-Winkler [10] are one of those edit distance based algorithms. Jaro-Winkler algorithm is a string matching algorithm that uses prefix-scale which makes it more accurate. It is the modified and extended method of Jaro Distance [11].

Here in (1), D_{Jaro} is Jaro distance, m is matched character's number that appeared in both spellings, t denotes the 'Number of Transpositions/2', |s1| and |s2| is 1st and 2nd string's length. In (2), $D_{Jaro\text{-}Winkler}$ is for the Jaro-Winkler distance, l denotes the common prefix length at the beginning of the word (limited to having 4 characters at maximum), p is the constant balancing element which decides how much the rating for particular prefixes is set up. p had a generic value of 0.1 in Winkler's original work [12].

$$D_{jaro} = \frac{1}{2} + \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \tag{1}$$

$$D_{jaro-winker} = D_{jaro} + \left(lp\left(1 - D_{jaro}\right)\right) \tag{2}$$

b. Token based; token based string similarity algorithm takes the words as a token and matches with the other token to get similarity percentage.

c. Sequence based; in sequence based string similarity algorithm, it goes for the largest common character set which is matched in both strings. The process is recursive and stop when no common sub string is found.

Table 2 shows the comparison of various string similarity algorithms to be "the" or "that" over the misspelled word "tha".

Table 2. String similarity distance comparison

String similarity algorithm	Distance	
	"the"	"that"
Hamming Normalized Similarity	0.667	0.75
Levenshtein Normalized Similarity	0.667	0.75
Jaro-Winkler	0.777	0.916
Jaccard index [13]	0.5	0.75
Sorensen-Dice	0.666	0.857
Ratcliff-Obershelp similarity [14]	0.666	0.857

3. RELATED WORKS

In the proposed approach by Khan *et al.* they worked with the phonetic encoding by Soundex algorithm [15]. At 2003, Abdullah *et al.* worked with direct dictionary searching process and recursive simulation algorithm for detecting typographic and cognitive phonetic errors and giving suggestions for misspelled words [16]. In the work of Z. Islam *et al.* at 2010 they applied stemming and edit distance algorithm [17]. First the inputted word is stemmed by removing only the suffixes from the word. If the stem is not correct, a suggestion generation procedure produces a list of suggestions. Among the suggested words the edit distance algorithm finds the best matched word. They achieved 90.8% accuracy in single error correction and for multiple error correction the rate is 67%, tested with 13,000 input words.

UzZaman *et al.* [18] tested with 1607 words and got 98% 1-error correction accuracy and 100% accuracy for 2-error correction. They used an immediate lexicon search technique for detection of an incorrectly spelled word. They made use of the patterns of error in normal writing to generate the correct spelling recommendations for an incorrectly spelled word. They also considered the patterns of phonetic error typically seen in Bangla writing. For gener-ating suggestions for typographic errors, they calculated edit distance between misspelled word and candidate words. For generating suggestions for phonetic errors, they used double metaphone encryption [19]. Finally, they considered each of the scores that were found by phonetic error and typographical error for indexing the suggestion list.

In the year of 2014, Chaudhuri [7] made a dictionary of phonetically alike characters. He made a single unit character code for mapping those indistinguishable characters. Another reversed dictionary is used and using string matching algorithm he found the phonetic errors. As it works with the phonetic similarity, it can only correct one error. Khan *et al.* [20] worked on the work of Munshi *et al.* [21] for evaluation of their system where they made use of 50,000 correct and 50,000 incorrect word by using N-Gram Model. Their outcomes are shown in Figure 1.

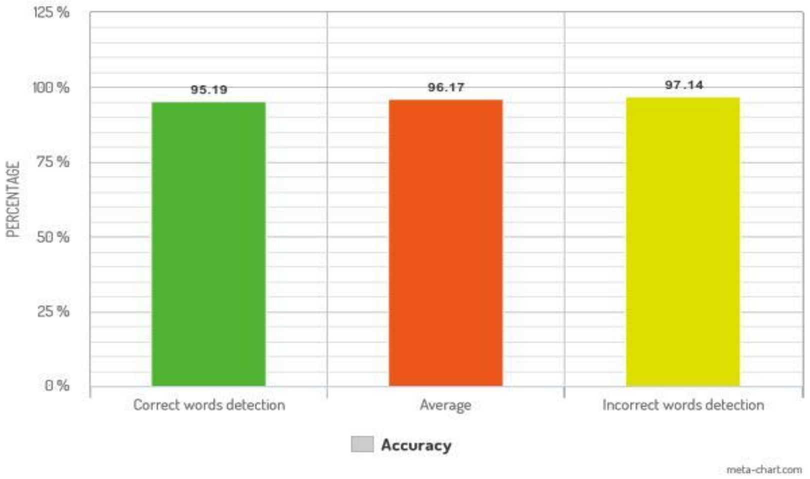


Figure 1. Evaluation result of [20]

The work done by Kumar *et al*. [22] showed the types of errors, error detection approaches and error correction techniques. They have done this study in context with other spell checker of Indian Languages. Etoori *et al*. [23] proposed based a character sequence-to-sequence text correction model for Hindi and Telegu Languages where they used LSTM encoder and decoder. For testing and evaluating they also build their own dataset. When measured the performance of their proposed system over other existing approaches, they got the highest accuracy of 85.4% whereas others have the 77.6% in case of Hindi Language. Jain *et al*. [24] proposed a method of detecting single word OOV or real word error where consists three main steps. First the data was collected in a confusion matrix which was used to explore frequency and types of error that had been occurred. Then using edit distance or predefined phonetically similar words are used to generate a candidate list and lastly correcting the sentence through Viterbi algorithm. It had the accuracy of 86% when threshold was greater than 5.

4. MATERIALS AND METHODS

In this section, our proposed process and material that we have used are briefly described.

4.1. Dataset

A huge collection of Bangla character combinations along with single characters is collected from [25], where 14980 individual character is placed. And also 959232 unique words are collected. We need this huge data set because if the corpus is enriched, the output will be more accurate.

4.2. Process

First, the system needs to take a Bangla word as input. The word may be correct or misspelled. Then it looks into the dictionary to find the word, if the word is found in the corpus the system will declare the input word as a correctly spelled word and terminate the process for this word. But if the word can not be found in the existing dictionary, the system will try to generate a list of suggestions for the correct word. The system will go for the 1-character edit first and then for the 2-character edit. In both cases, the word will be split by one character and based on those splits there will be deletion of n-character(s), insertion of n-character(s), Transpose within n-character(s) and replace n-character(s) with new n-character(s) from the data set of individual characters according to the split sets outcome where n can be 1 or 2, and then it will generate a long list containing all outputs of all steps mentioned above. The list now contains some correct words and a huge number of incorrect words.

The list of words are then matched with the dictionary words and correct words are shown as suggestions for that misspelled word. Among those suggestions the most probable one chose by index number. Index number is generated by distance of string similarity algorithms. The best result is found by applying Jaro-Wrinkler distance algorithm discussed in section 2.4 (a). The lower the distance, the greater the probability of higher index. The highest index that represents a word from the final list then becomes the most probable word. The whole process is shown in Figure 2.

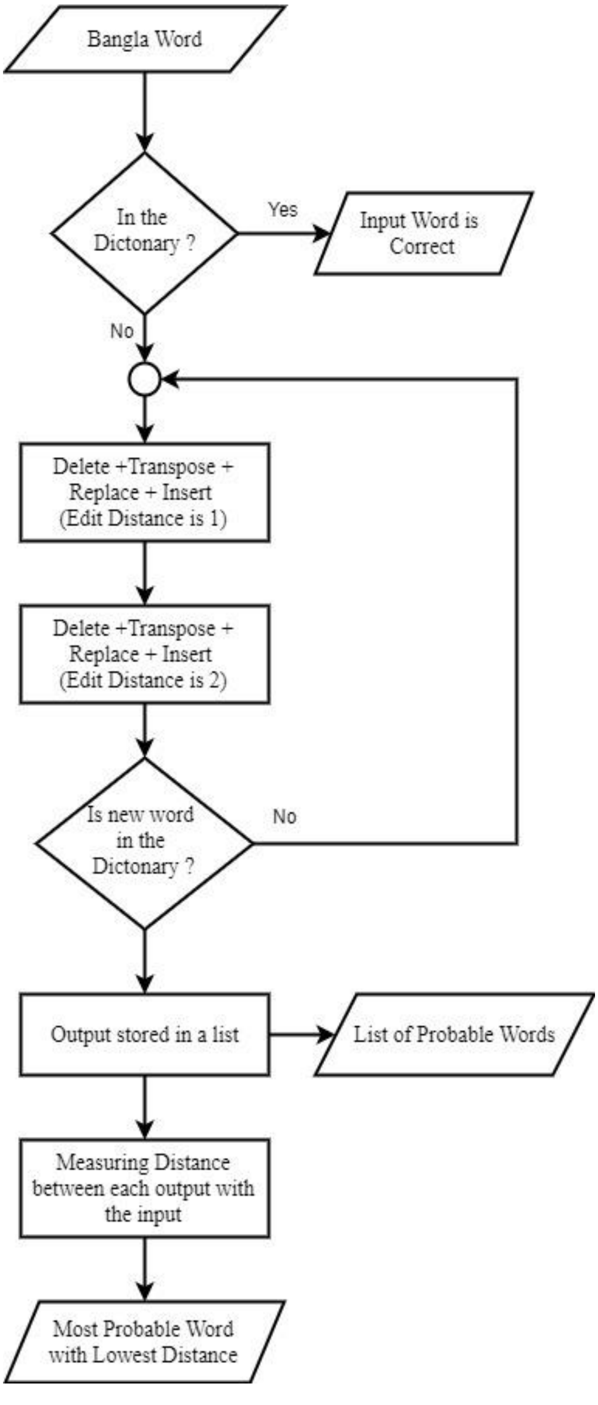

Bangla Word
In the Dictonary ?
Yes
Input Word is Correct
No
Delete +Transpose + Replace + Insert (Edit Distance is 1)
Delete +Transpose + Replace + Insert (Edit Distance is 2)
Is new word in the Dictonary ?
No
Output stored in a list
List of Probable Words
Measuring Distance between each output with the input
Most Probable Word with Lowest Distance

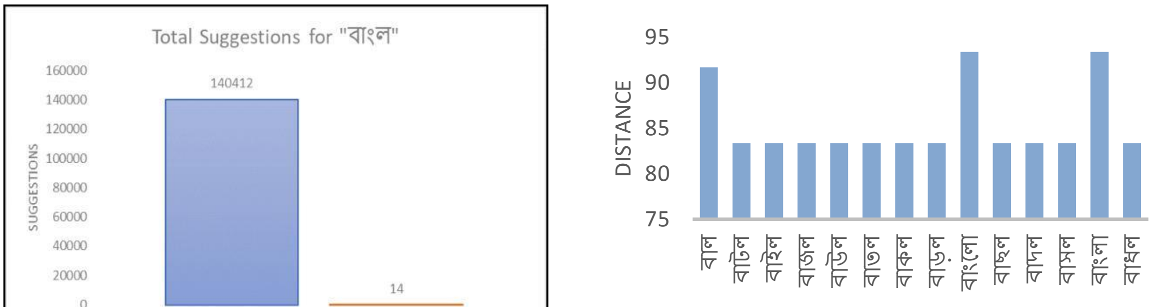
Figure 2. Flowchart of our proposed method for correcting Bangla language

5. RESULTS AND DISCUSSION

In this proposed approach we used string similarity as a factor to generate a list of suggested words and picking aword as the correct one. But our system does not make use on any probability measures. Adding the usage of occurrence probability scores from a huge corpus of Bangla Literature with our system will make the final output more accurate according to the Algorithm 1. We focused on the non-word errors in this work. So another limitation of our work is not considering real-word errors. Another limitation of our work is that it is not as fast as some of the spell correcting algorithms available in other languages specially in English. Our algorithm is also highly dependant on the dictionary that is used but it is an unavoidable factor when it comes to spelling correction.

```
Algorithm 1: Most Accurate Bangla Word Selection from a list of correct word suggestion
Input : Suggestion Set s
Output: String r
Data: Testing set x

/* Testing set can be a book or a newspaper orany other site which contains a large
numberof Bangla articles. */

for words in x do
      Split the article by " " (space)
      word[i]←each word
      count[word[i]]←similar words
      P (word[i])←count[word[i]]=length(word)
end
for words in s do
      $P_s$[i]←P(word[i])
      if $P_s$[i] is MAX then

/* Maximum Probability of a word is checked among the list's words by putting the valuein
a temporaryvariable.*/

      r ← s[i]
      end
end
```

We have tested over 1000 Bangla words, and it detected 970 words as correct word or misspelled word and gave proper suggestions' list in case of misspelled one. Those 1000 Bangla words are collected randomly from different comments of various Facebook pages. If we would take consideration of the word mentioned in Figure 3 (a), we could see that the system showed it as misspelled word and gave a list of probable words where the list consists of 14 correctly spelled words. Those are *'বাধল'*, *'বাংলো'*, *'বাড়ল'*, *'বাংলা'*, *'বাতল'*, *'বাল'*, *'বাইল'*, *'বাসল'*, *'বাদল'*, *'বাজল'*, *'বাটল'*, *'বাউল'*, *'বাছল'*, *'বাকল'*. In Figure 3 (b) we got the comparison of those suggestions and found the distance between suggested word and misspelled word in terms of percentage.



(a) (b)

Figure 3. Quantity of suggested words; (a) total suggestion for the misspelled Bangla word, (b) suggested word with distance (in %) with the misspelled word

This system was compatible to detect an error which was out of vocabulary. In case of choosing the most probable word as it uses the edit distance so there was a huge probability of having the highest and at the same time that edit distance might be of more than one word. In that case the last word found was taken but this process needed to be modified by having contextual perspective over the whole sentence. As we discussed in the section 4.1, the performance of our system depends on the size of the corpus. If we can enlarge the corpus the output suggestions will be more accurate. A comparison between the performance of previous works and our work is shown in Table 3.

Table 3. Comparison between existing works and our proposed method

Existing related works (author)	Method/Approach	Performance (accuracy)
Bidyut Baran Chaudhuri [20]	String matching algorithm	Tested over 25000 words, they found 5% false positive detection, but this approach needs twice the memory space to hold one dictionary with its reverse dictionary.
Naushad Uz Zaman and Mumit Khan.[15]	Double Metaphonic encoding	91.37% accurate result tested over 1607 words
Z. Islam, N. Uddin, and Mumit Khan.[17]	Stemming algorithm and Edit distance	90.8% and 67% respectively for correcting 1-error and multiple error misspellings tested over 13000 words but it can't handle derivational suffixes.
N. Uz Zaman and M. Khan [19]	Direct dictionary looks up method, Double Metaphonic encoding and Edit distance	Tested over 1607 words, 98% and 100% respectively for correcting single error and 2-error misspellings. But this process creates a lot of suggestions for detectable multiple errors.
Md Munshi Abdullah, Md Zahurul Islam and Mumit Khan.[21]	Finite state automaton	92% and 70% respectively for correcting single character and multiple character misspellings tested over 291 words. But it cannot deal with transposition errors as an error of single edit distance
Our Proposed System	Norvig's Algorithm and Jaro-Winker Edit Distance	97% accurate in both cases of single and multiple error handling

6. FUTURE WORK

As future work, we will expand the scope of our work by including the correction of real-word error by adding pattern matching. This system predicts the most probable word from the suggestion list based on JW Distance. But to achieve the accurate context of the text the suggested word may not appropriate. It can be solved by having a probability of word occurrence in that document by TF-IDF and that probability will be considered along with the edit distance value to predict the most accurate word from the suggested list generated through Norvig's algorithm. Again, the dictionary that was used can be more enriched by analyzing different social media's posts and comments of Bangla language and from there an occurrence probability will be calculated for a specific word and will be stored in the dictionary along with each word. When this dictionary will be used it will also match that word's occurrence probability in the internet also in the specified document from where the misspelled word had been picked.

7. CONCLUSION

Spelling mistakes might not be a recent phenomenon, but with the increase in the usage of social media and micro-blogging websites, it sure is more prevalent in recent years. While there are abundant instances of works done in this field of spelling mistake detection and correction, the number of works done in Bangla language are not more than a few. Our work presents an approach to detect spelling mistakes, prepare a list of suggestions for the correct word and to choose a word from the list as the correctly spelled word for the input word. For this, we used Norvig's algorithm along with Jaro-Winklers distance as a measure of string similarity. Our method offered 97% accuracy when tested 1000 misspelled words.

ACKNOWLEDGEMENTS

The author would like to thank Bangladesh University of Professionals. This work has been supported by Department of Information and Communication Technology under Assistant Professor Dr. S M Salim Reza.

REFERENCES

[1] P. Norvig, "How to write a spelling cor-rector," [Online]. Available at: http://norvig.com/spell-correct.html. Feb. 2007.
[2] S. K. Chatterji, "Bengali Phonetics," *Bulletin of the School of Oriental Studies, University of London*, vol. 2, no. 1, pp. 1–25, 1921, doi:10.1017/S0041977X0010179X.
[3] K. L. Čižikova, and C. A. Ferguson, "Bibliographical Review of Bengali Studies," *Linguistics in South Asia*, pp. 85-98, 2016, doi: 10.1515/9783110819502-007.
[4] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, vol. 24, no. 4, pp. 377-439, 1992, doi: 10.1145/146370.146380.
[5] X. Tong and D. A. Evans. "A statistical approach to automatic OCR error correction in context," *Fourth Workshop on Very Large Corpora*, 1996.

[6] G. Hirst, A. Budanitsky, “Correcting real-word spelling errors by restoring lexical cohesion,” *Natural Language Engineering*, vol. 11. no. 1, pp. 87-111, 2005.

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

[8] M. Gilleland, M. P. Software, “Levenshtein distance, in three flavors”, [Online]. Available at: https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein%20Distance.htm.

[9] T. Ahmed, “String: trigram - find similar strings by tri-gram (or 1, 2, 4, etc.-gram) method,” [Online]. Available at: https://metacpan.org/pod/distribution/String- Trigram/ Trigram.pm.

[10] Y. Wang, J. Qin, and W. Wang, “Efficient approximate entity matching using jaro-winkler distance,” *Web Information Systems Engineering-18th International Conference*, pp. 231-239, 2017.

[11] M. A. Jaro, “Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida,” *Journal of the American Statistical Association*, vol. 84:406, pp. 414-420, 1989, doi: 10.1080/01621459.1989.10478785.

[12] W. E. Winkler, “String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage,” *American Statistical Association 1990 Proc.s of the Section on Survey Research Methods*, pp. 354-359, 1990.

[13] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, “Using of Jaccard coefficient for keywords similarity,” *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, no. 6, pp. 380-384, 2013.

[14] J. W. Ratcliff and D. E. Metzener, “Pattern-matching-the gestalt approach,” *Dr Dobbs Journal*, vol. 13, no. 7, p. 46, 1988.

[15] M. Khan and N. UzZaman, “A Bangla phonetic encoding for better spelling suggestion,” *Center for Research on Bangla Language Processing*, pp. 76-80, Jan. 2004.

[16] A. B. A. Abdullah, A. Rahman, “A generic spell checker engine for south Asian languages,” *Conference on Software Engineering and Applications*, pp. 3-5, 2003.

[17] Z. Islam, N. Uddin and M. Khan, “A light weight stemmer for bengali and its use in spelling checker,” *Center for Research on Bangla Language Processing*, 2007.

[18] N. UzZaman and M. Khan, “A comprehensive bangla spelling checker,” *Center for Research on Bangla Language Processing*, 2006.

[19] N. UzZaman and M. Khan, "A Double Metaphone encoding for Bangla and its application in spelling checker," *2005 International Conference on Natural Language Processing and Knowledge Engineering*, 2005, pp. 705-710, doi: 10.1109/NLPKE.2005.1598827.

[20] N. Hossain Khan, G. C. Saha, B. Sarker and M. H. Rahman, “Checking the correctness of bangla words using n-gram,” *International Journal of Computer Applications*, vol. 89, no. 11, pp. 1-3, 2014, doi: 10.5120/15672-4416.

[21] M. Munshi, M. Khan, “Error-tolerant finite-state recognizer and string pattern similarity based spelling-checker for bangla,” Sep. 2010.

[22] R. Kumar, M. Bala, and K. Sourabh, “A study of spell checking techniques for Indian languages,” *JK Research Journal in Mathematics and Computer Sciences*, vol. 1, no. 1, pp. 105-113, 2018.

[23] P. Etoori, M. Chinnakotla, and R. Mamidi, “Automatic spelling correction for resource-scarce languages using deep learning,” *Proceedings of ACL 2018, Student Research Workshop*, pp. 146-152. 2018.

[24] A. Jain, M. Jain, G. Jain, and D. K. Tayal, “"UTTAM”: An efficient spelling correction system for hindi language based on supervised learning,” *ACM Transactions Asian Low-Resource Language Information Process*, vol. 18, no. 1, pp. 1-26, 2018, doi: 10.1145/3264620.

[25] Bengali character combinations, [Online]. Available at: https://github.com/MinhasKamal/BengaliDictionary. [Accessed: 2019-10-26].

BIOGRAPHIES OF AUTHORS

Istiak Ahamed is curreenty working as a software engineer at Shuttle Technologies BD Ltd. He received his degree in Information and Communication Engineering from Bangladesh University of Professionals, Dhaka, Bangladesh, in 2020. Till now he has published one conference paper. His research interest in Artificial Intelligence, Machine Learning, and application of different ML Technologies.

Maliha Jahan is working as a Lecturer in the Department of Communication Science and Engineering (CSE) at United International University. She completed her B.Sc. degree in Information and Communication Engineering (ICE) from Bangladesh University of Professionals (BUP), Dhaka, Bangladesh in 2020. She is enrolled in the Electrical and Computer Engineering (ECE) Ph.D. program at Johns Hopkins University. She has previously worked on one international journal paper and one international conference paper.

Zarin Tasnim is working as a Lecturer in the Department of Information and Communication Technology (ICT) under the Faculty of Science and Technology (FST), Bangladesh University of Professionals (BUP). She achieved B.Sc. (Engineering) degree in Electrical, Electronic and Communication (EECE) Engineering from Military Institute of Science and Technology (MIST), Dhaka, Bangladesh in 2014 and M.Sc. (Engineering) degree in Telecommunications and Wireless Systems from the University of Liverpool, UK in 2019. She has one national journal and one international conference publications.

Tajbia Karim is serving as a lecturer in Information and Communication Technology (ICT) department at Bangladesh University of Professionals (BUP). Previously she worked in Electrical and Electronic Engineering department of American International University-Bangladesh. Besides being an academician, she is voluntarily working as a student advisor of the ICT department students enrolled in 2019 at BUP. Till now she has published two journal and three conference papers. She has a diversified research interest including Electronics, Signal processing, Data science etc.

S. M. Salim Reza is an Assistant Professor of the Department of ICT, Bangladesh University of Professionals (BUP), Dhaka, Bangladesh. He is a Professional Member of IEEE and many other professional organizations. He is a keen planner & implementer with proficiency in managing academic and administrative activities entailing operational policies/norms, faculty appraisal/ computer science training. He has more than 50 research papers in ISI/Scopus Indexed journal and many reputed International conferences.

Dilshad Ara Hossain has graduated in Computer Science and Engineering from Dhaka, Bangladesh. Currently, she has been studying in her final stage of Masters (by research) in Department of Information and Communication Technology (ICT), in International Islamic University Malaysia, Kuala Lumpur, Malaysia. She has many Indexed Journal paper and renowned conference proceedings. Her research interest is in Cyber Security, IoT, Artificial Intelligence, Machine Learningm, VANET and recent computer applications.