I.J. Intelligent Systems and Applications, **2018, 2, 47-54**
Published Online February 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2018.02.05

An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction

Md. Tarek Habib
Department of CSE, Daffodil International University, Dhaka, Bangladesh
E-mail: md.tarekhabib@yahoo.com

Abdullah Al-Mamun, Md. Sadekur Rahman, Shah Md. Tanvir Siddiquee
Department of CSE, Daffodil International University, Dhaka, Bangladesh
E-mail: mamun.education@gmail.com, sadekur738@gmail.com, tanvir.cse@diu.edu.bd

Farruk Ahmed
Department of CSE, Independent University Bangladesh, Dhaka, Bangladesh
E-mail: farruk60@gmail.com

Received: 16 April 2017; Accepted: 15 September 2017; Published: 08 February 2018

***Abstract*—**Word completion and word prediction are two important phenomena in typing that have intense effect on aiding disable people and students while using keyboard or other similar devices. Such auto completion technique also helps students significantly during learning process through constructing proper keywords during web searching. A lot of works are conducted for English language, but for Bangla, it is still very inadequate as well as the metrics used for performance computation is not rigorous yet. Bangla is one of the mostly spoken languages (3.05% of world population) and ranked as seventh among all the languages in the world. In this paper, word prediction on Bangla sentence by using stochastic, i.e. *N*-gram based language models are proposed for auto completing a sentence by predicting a set of words rather than a single word, which was done in previous work. A novel approach is proposed in order to find the optimum language model based on performance metric. In addition, for finding out better performance, a large Bangla corpus of different word types is used.

***Index Terms*—**Word prediction, performance metric, natural language processing, *N*-gram, language model, corpus, machine learning, eager learning.

I. Introduction

Writing and typing aids for the disabled are so important. A person having disability can live a comfortable life if he or she has the opportunity of leaving a note or typing an email easily being aided through automatic sentence completion technique by the process of word prediction. In addition, for the early learners in any field (i.e. students, novice researchers) the automatic sentence completion technique could be beneficial during the learning process by searching new things; as word prediction might help them by providing most suitable suggestions while searching for new topics with keywords. Besides, word prediction is an "intelligent" word processing feature that can alleviate writing breakdowns simply by reducing the number of keystrokes necessary for typing words, as with the inputs of a letter, the software presents a list of possible words beginning with that letter. As each additional letter is added, the list is refined. When the intended word appears in the list, the person selects it, often by clicking on it or typing its number, which inserts the word into the document. This requires a rigorous performance metric rather than traditional performance metric, i.e. accuracy which only considers whether the predicted word matches with the intended word. Auto-completion has been extensively implemented in most modern text editors, browsers and search engines. In predictive auto-completion systems, the candidates are matched against the prefix on-the-fly using information retrieval and natural language processing (NLP) techniques. NLP not only encompasses the problems related to text, such as text documents classification [1], text translation [2], etc., but also the problems related to speech, such as [3]. In recent years, machine learning is widely being practiced in NLP. In this paper, a novel approach is proposed in order to find the optimum language model based on a novel performance metric for automatic sentence completion using supervised machine learning technique based on popular *N*-gram language model for Bangla language.

The rest of the paper is organized as follows. Section II describes the current state of solution to address the problem of automated Bangla word prediction. Section III describes the formal problem formulation of the

automated Bangla word prediction. In Section IV, comes the description of our approach to solve the problem. Section V describes how we apply our entire methodology and what results are achieved. In Section VI, we investigate results obtained in order to develop an understanding about the merits of our proposed approach. Finally, we summarize our work along with limitations, and discuss the scope for future work in Section VII.

II. Related Works

Very few efforts have been made for word prediction for a recent couple of years, specially focused on the language Bangla. In the previous work in Bangla [4], word prediction on sentence by using stochastic, i.e. *N*-gram based language models such as unigram, bigram, trigram, linear interpolation and backoff models are proposed for auto completing a sentence by predicting a single word in a sentence which is different than the work presented in this paper because now the prediction is built with a set of words instead of a single one during finding out the best language model. In [5], the authors developed a sentence completion method in both German and English based on *N*-gram language models and they derived a k best Viterbi beam search decoder for strongly completing a sentence. The use of Artificial Intelligence for word prediction in Spanish is also observed in [6], in which using the chart bottom-up technique, syntactic and semantic analysis is done for word prediction. In [7], an effective method of word prediction in English is presented using machine learning and new feature extraction and selection techniques adapted from Mutual Information (MI) and Chi square (χ^2). Nagalaviand and Hanumanthappa [8] have applied *N*-gram based word prediction model in order to establish the link between different blocks of a piece of writing in e-newspaper in English retaining with the sentence reading order. Some researchers use *N*-gram language model for word completion in Urdu language [9] and in Hindi language [10] for detecting disambiguation in Hindi word. Some related works in Bangla language, e.g. Bangla grammar checker [11] using *N*-gram language model, checking the correctness of Bangla word [12], verification of Bangla sentence structure [13], and validity determination of Bangla sentences [14] are also conducted. There are different word prediction tools such as AutoComplete by Microsoft, AutoFill by Google Chrome, TypingAid, LetMeType etc. Some software developed to provide only word completion features but do not offer word prediction or sentence completion features.

At [15] software with improved training and recall algorithms are suggested to solve the sentence completion problem using the cogent confabulation model, which can remember sentences with 100% accuracy in the training files. In addition, it helps in filling up missing words in simple sentences or based on some given initial words deliver meaningful sentences. In [16], a *N*-gram model is constructed which was used to compute 30 alternative words for a given low frequency word in a sentence, and human judges then picked the 4 best impostor words, based on a set of provided guidelines. They also used the CMU language modeling toolkit to build a 4-gram language model using Good-Turing smoothing. However, during their experiment though the human grooming shows good accuracy (91%) but the *N*-gram models results(Generating model-31%, Smoothed 3-gram- 36%, Smoothed 4-gram-39%, Simple 4-gram-34% and LSA similarity model-49%) do not actually meet the expectation.

For sentence-completion task an index-based retrieval algorithm and a cluster-based approach are proposed at [17]. Bickel et al. [18] learned a linearly interpolated *N*-gram model for sentence completion. For generating and ranking auto-completion candidates for partial queries in the absence of search logs, Bhatia et al. [19] extracted frequently occurring phrases and *N*-grams from text collections and deployed them. For the purpose of learning to personalize auto-completion rankings based on a new approach is proposed in [20]. While previous auto-completion models rely on aggregated statistics across all (or demo-graphic groups of) users, they showed that user-specific and demographic-based features can be used together under the same framework. They also introduced a strategy for extracting training labels from previous logs and showed that it can be used for training auto-completion rankers. Though they considered their labels to be binary, it would be interesting to investigate how multi-graded labels. For the future work they left their model, which can be extended by allowing more than one relevant candidate per ranking if they are closely related. However, the word prediction process using *N*-gram language model presented in this paper by shows significantly more accuracy than the above mentioned works. In addition, during the word prediction for the sentence completion process a novel approach is followed in which a set of words (collected and sorted based on a ranking mechanism) is used as list of suggestions for sentence completion instead of single word (binary prediction).

III. Problem Description

The problem addressed in this paper is about stochastically predicting a suitable word to complete an incomplete sentence which consists of some words. Let $w_1 w_2\ w_3\ \ldots\ w_{m-1} w_m$ be a sentence (i.e. sequence of words) where $w_1 w_2\ w_3\ \ldots\ w_{m-1}$has already been typed. The problem of the task is to build a language model which takes $w_1\ w_2\ w_3\ \ldots\ w_{m-1}$ as input and predicts an *n*-tuple of words (v_{m1},v_{m2}, v_{m3}, $\ldots v_{mn}$) as output in order to match an element (possibly first) of the *n*-tuple with w_m(intended word), as shown in Fig. 1.

The performance of each language model is measured by taking both the matching of predicted word with intended word as well as the order of matching into account. Therefore, accuracy and failure rate are used in order to address this issue.

Suppose w_m matches with v_{mi}, (i.e. w_m equals v_{mi}, where $1 \leq i \leq n + 1$), then the equation of accuracy is as follows:

$$Acuracy = \frac{n+1-i}{n} \times 100\% \quad (1)$$

Failure occurs when i equals n + 1. (n + 1)-th match means no match has taken place, i.e. accuracy equals 0. If in an experiment a language model fails to predict f times, i.e. f failures occur, out of p predictions, then the failure rate is likewise:

$$Failure\ Rate = \frac{f}{p} \times 100\% \quad (2)$$

Another aspect of the problem is empirical. Given a number of language models, we need to come up with the one which outperforms all other models in terms of accuracy for possibly small value of n.

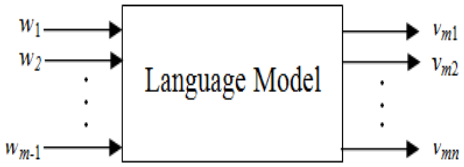

Fig.1. Proposed model.

IV. Proposed Approach

We begin with five language models, namely unigram, bigram, trigram, backoff and linear interpolation. All these language models are based on N-gram approximation. The general equation for this N-gram approximation to the conditional probability of the next word in a sequence is:

$$P\left(w_n \mid w_1^{n-1}\right) \approx P\left(w_n \mid w_{n-N+1}^{n-1}\right) \quad (3)$$

Equation 3 shows that the probability of a word w_n given all the previous words can be approximated by the probability given only the previous N words.

We train a language model based on a corpus setting n, the prediction length, with 1. Then accuracy of the trained model is tested. The value of n is increased by 1 and the language model is trained and tested. The process continues until insignificant change in accuracy occurs and the value exceeds the average word length of corpus, $|\overline{w}|$. Here is to mention that as the value of n increases, so is for accuracy too. Although larger value of n involves better accuracy, it increases the value of i, the number of position in n-tuple at which prediction matches. Thus it also involves larger number of key strokes required. This is why the average word length of corpus is used in looping condition. In this way, n*, the considerable optimum value of n is automatically calculated for every language model stated earlier, which is given as pseudocode in Algorithm 1.

In Algorithm 1 "considerable optimum value of n" means that the minimum value of n, for which the accuracy is the maximum among the accuracies for all values of n throughout the range of for loop iteration in Algorithm 1.

After computing n* as well as accuracy for all language models, the maximum accuracy is calculated. The language model, which shows the maximum accuracy, is selected as optimum model. If there are more than one language models, which show the same or almost same accuracy, the language model with the minimum failure rate is selected as the optimum model. If there are more than one language models with the minimum or near about minimum failure rate, the language model, which corresponds with the most positively skewed curve for the accuracy distribution among different values of i-th match ($1 \leq i \leq n$*), is selected as the optimum language model. The entire methodology is shown in Fig. 2.

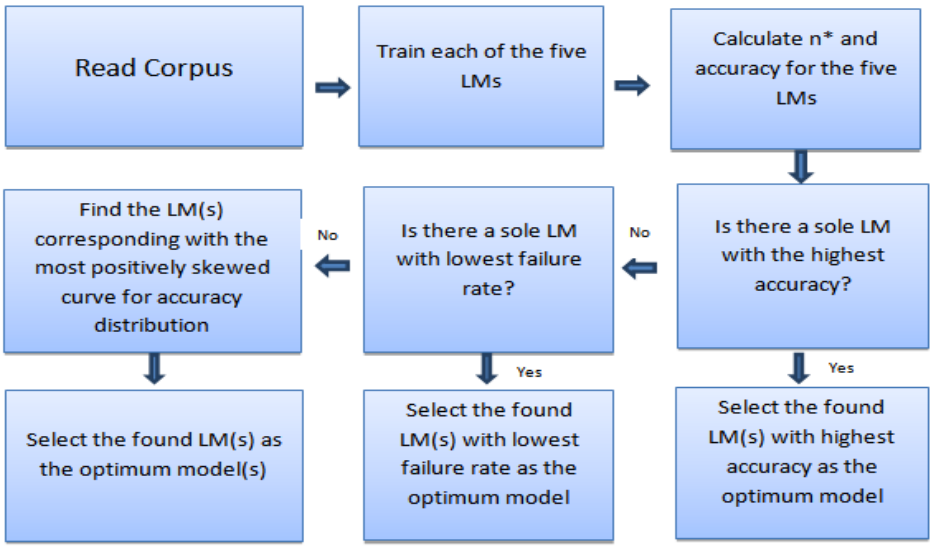

Fig.2. Proposed process offinding best language model (LM).

Algorithm 1. Algorithm for calculating *n**, the considerable optimum value of *n* for each language model (LM), where ε is a small positive number working as the stopping criterion and $|\bar{w}|$ is the average word length of corpus.

```
Set n* to 0 and the value of minimum accuracy to accuracy of LM using n = 1
for i ← 2 to |W̄|
   Calculate the accuracy of LM using n = i
   if current accuracy < minimum accuracy
      Set the value of minimum accuracy to current accuracy
   Set δ to the nonnegative difference of current and previous accuracy
   if δ ≤ ε
      Set n* to the value of i, for which accuracy is minimum
Exit for loop
```

V. Implementation

A set of training modules of word prediction were developed to compute unigram, bigram, trigram, backoff as well as linear interpolation based on *N*-gram. The implementation is different in respect to the previous work [4] as the prediction is built with a set of words instead of a single one during finding out the best language model among these language models. These models are used to determine different probabilities by counting frequencies of words in a very large corpus, which has been constructed from the popular Bangla newspaper the "Daily Prothom Alo". The corpus contains more than 11 million (11,203,790) words and about 1 million (937,349) sentences, where total number of unique words is 294,371 and average word length ($|\bar{w}|$) is 7.

During this work, the entire corpus is divided into two parts, namely training part and testing part. The holdout method [21] is used to split the corpus at the proportion of two-thirds for training and one-third for testing. Therefore, this work starts with a training corpus of size more than six (6) hundred thousand sentences. In order to avoid model over-fitting problem (i.e. to have lower training error but higher generalization error), a validation set is used. In accordance with this approach, the original training data is divided into two smaller subsets. One of the subsets is used for training, while the other one (i.e. the validation set) is used for calculating the generalization error. Two-thirds of the training set is fixed for model building while the remaining one-third is used for error estimation.

The holdout method is repeated for five times in order to find the best model. After finding out the best model, the accuracy of the model is computed using the test set, through which the considerable optimum prediction length (*n**) is determined automatically based on Algorithm 1. The optimum prediction length (*n**) along with the accuracy of each model is shown on Table 1.

From the Table 1, it can be seen that the optimum prediction length (*n**) is seven for all of the models except the bigram (*n**=5). The accuracy comparison of all the models is presented in Fig. 3, where the optimum prediction length (*n**) of each model is also marked with yellow dot.

Table 1. Optimum prediction length (*n**) of all language models

Language Model	Prediction Length							Optimum value of *n* (*n**)
	n = 1	*n* = 2	*n* = 3	*n* = 4	*n* = 5	*n* = 6	*n* = 7	
Unigram	2.70%	11%	19.50%	22%	28.50%	32.50%	34%	7
Bigram	35%	39%	45.50%	48.33%	55%	50%	50%	5
Trigram	59.50%	66%	67%	68.50%	70%	74.50%	75%	7
Backoff	60%	66.10%	67%	68%	71%	74.50%	75.90%	7
Linear Interpolation	61%	66.50%	67.50%	69%	71.50%	75.60%	77%	7

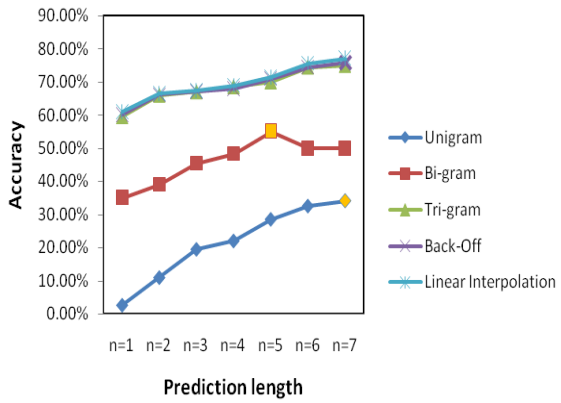

Fig.3. All models' accuracy comparison against the prediction size.

In addition, a detailed investigation is conducted (shown on Table 2) to evaluate the performance of the classifier for all models by varying the length of test sentences, i.e. unigram, bigram, trigram, backoff and linear interpolation.

The comparison of the top three model's accuracy against the average availability of the words in the test sentences is shown on Fig. 4.

After finding out the different accuracy rate of top three model's with the test set consists of sentences with different lengths, the average accuracy of the model's (i.e. trigram, backoff and linear interpolation) is computed (see Fig. 5) which might lead us in finding out the best language model. During finding out the accuracy of each model, it is noticed that, sometimes models show almost

same accuracy during the process of predicting the suitable word. Therefore, keeping track of the failure rate is considered as a significant task, as some models might show same accuracy but with different failure rate. In Table 3, the failure rate of all the models is presented.

Table 2. All models' accuracy across the availability of words

Available Words in Test Sentences	Accuracy of Language Model		
	Trigram	Backoff	Linear Interpolation
1	39.00%	40.25%	40.5%
2	46.30%	47.30%	47.4%
3	48.60%	49.00%	49%
4	58.00%	58.50%	58.8%
5	60.90%	59.30%	59.5%
6	62.50%	62.70%	62%
7	65.70%	66.60%	66.7%
8	67.27%	67.50%	67%
9	69.60%	69.90%	70.2%
10	71.50%	70.20%	70.5%
11	72.35%	72.50%	71%
12	75.50%	75.63%	75.63%
13	78.00%	78.50%	79%
14	80.50%	80.70%	80.9%
15	81.15%	81.40%	81.5%

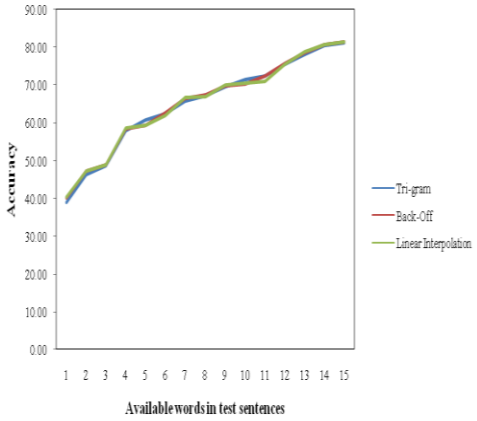

Fig.4. All models' accuracy comparison against the availability of words.

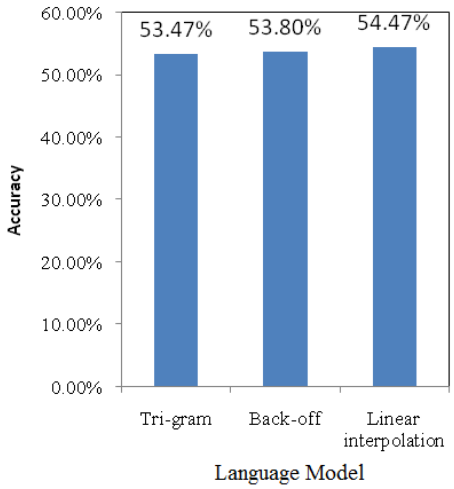

Fig.5. Average accuracy of language models.

Table 3. All models' failure rate with the availability of the words in test sentences

Available Words in Test Sentences	Failure Rate of Language Model		
	Trigram	Backoff	Linear Interpolation
1	38.53%	40.15%	31%
2	33.23%	31.93%	30.3%
3	32.93%	30.23%	28%
4	28.03%	26.73%	25.5%
5	28.43%	25.53%	25%
6	26.93%	24.94%	23.9%
7	24.83%	23.09%	21.5%
8	23.23%	21.33%	20.5%
9	22.93%	19.23%	18.43%
10	20.53%	16.65%	13.8%
11	16.17%	13.33%	10.67%
12	12.9%	12.2%	8.6%
13	9.5%	7.8%	6.15%
14	5%	4.3%	3.7%
15	2.3%	2.1%	2%

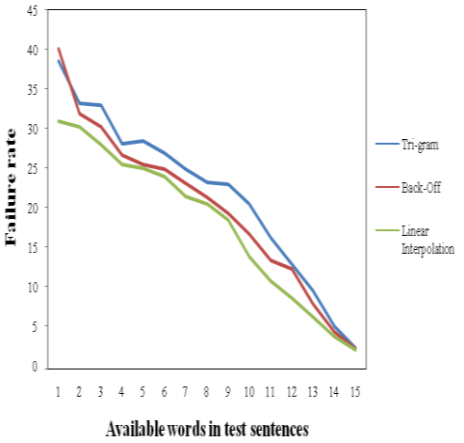

Fig.6. All models' failure rate comparison against the availability of words.

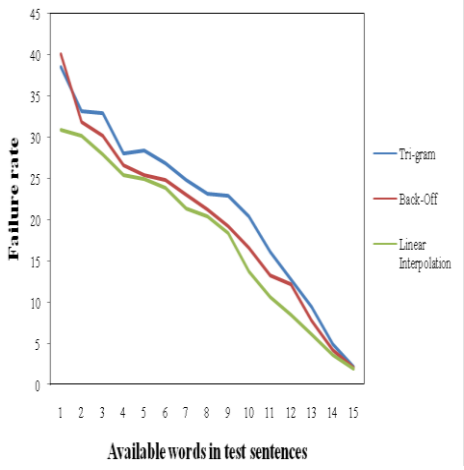

Fig.7. Average failure rate of language models.

The comparison of the top three model's failure rate against the average availability of the words in the test sentences is shown on Fig. 6.

After finding out the different failure rate of top three

model's with the test set consists of sentences with different lengths, the average failure rate (see Fig. 7) of the top model's (i.e. trigram, backoff and linear interpolation) is computed which might lead us in finding out the best language model; as during the process of finding out the accuracy some models have shown almost similar accuracy which makes the selection process difficult.

VI. Discussion

During the initial experiment, as shown on Table 1, it is noticeable, the top three models have shown good accuracy among all the models (see Fig. 3), though linear interpolation model shows slightly better performance in terms of predicting next possible word with optimum prediction length seven, i.e. $n^* = 7$. Therefore, to find out the best model, in the second phase, a further deep investigation is conducted, as shown on Table 2, to find out, how the top three models behave against the test sets with different sizes (average) of sentences. From the experiment in second phase, all the top models behave similar like before (see Fig. 4). Consequently, a third phase is required, in which the failure rate of the top models is computed (see Table 3). Though, in some cases the trigram, backoff and linear interpolation method show almost same accuracy, but the failure rate of the other two models (trigram and backoff) is higher compared to the linear interpolation (see Fig. 6). Moreover, from the average accuracy and average failure rate of all models (Fig. 5 and Fig. 7 respectively) it is obvious to come up with the final decision that linear interpolation model accomplishes most accuracy among all other models during the word prediction process. The accuracy rate along with the increment of the prediction length of the linear interpolation model is shown on Table 4 and Fig. 8.

Table 4. Accuracy along with the prediction length of Linear Interpolation model.

Prediction length (n)	Accuracy
1	61%
2	66.50%
3	67.50%
4	69%
5	71.50%
6	75.60%
7	77%

Although the linear interpolation model has shown better performance than other top models (77% with $n^* = 7$) and the experiment result is promising; it is still necessary to test with larger training corpus to accomplish more than 90% accuracy with less prediction length (n^*).

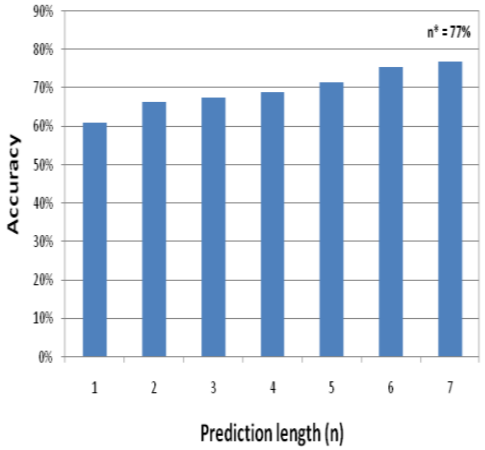

Fig.8. Accuracy along with the prediction length of Linear Interpolation model.

VII. Conclusion

The focus of this research was modeling, training and recall techniques for automatic sentence completion using supervised machine learning technique based on popular *N*-gram language model. *N*-gram based word prediction works well for English, but for Bangla language, it is found more challenging to get very good, e.g. more than 90% accuracy, performance as it depends on training corpus of size more than six hundred thousand sentences. Though during the several phases of experiments, the top three models show almost same level of accuracy, but in terms of both accuracy and failure rate, the linear interpolation outperforms the other models.

For the future work, a further testing with the present models is planned with larger corpus. In addition, it is perceived to find out a language model with a better algorithm that will make use of character level *N*-gram in combination with word level *N*-gram, or make use of Bangla grammatical rules along with word level *N*-gram. Extensive work is in progress to devise such a language model using a very large Bangla corpus.

References

[1] I. S. I. Abuhaiba and H. M. Dawoud, "Combining Different Approaches to Improve Arabic Text Documents Classification", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.9, No.4, pp.39-52, 2017.

[2] G. Chandra, S. K. Dwivedi, "Assessing Query Translation Quality Using Back Translation in Hindi-English CLIR," *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.9, No.3, pp.51-59, 2017.

[3] K. K. Ravi and P.V. Subbaiah, "A Survey on Speech Enhancement Methodologies," *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.8, No.12, pp.37-45, 2016.

[4] M. M. Haque, M. T Habib and M. M. Rahman, "Automated Word Prediction in Bangla Language Using Stochastic Language Models," *Academy & Industry Research Collaboration Center (AIRCC) International Journal in Foundations of Computer Science & Technology*, vol. 5, no. 6, pp. 67–75, November 2015.

[5] S. Bickel, P. Haider and T. Scheffer, (2005), "Predicting Sentences using *N*-Gram Language Models," *In Proceedings of Conference on Empirical Methods in Natural language Processing*.

[6] N. Garay-Vitoria and J. Gonzalez-Abascal, (2005), "Application of Artificial Intelligence Methods in a Word-Prediction Aid," Laboratory of Human-Computer Interaction for Special Needs.

[7] H. Al-Mubaid, "A Learning-Classification Based Approach for Word Prediction," *The International Arab Journal of Information Technology*, Vol. 4, No. 3, 2007.

[8] D. Nagalaviand and M. Hanumanthappa, "*N*-gram Word prediction language models to identify the sequence of article blocks in English e-newspapers," *In Proceedings of International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2016.

[9] Q. Abbas, (2014), "A Stochastic Prediction Interface for Urdu", *Intelligent Systems and Applications*, Vol.7, No.1, pp 94-100 .

[10] U. P. Singh, V. Goyal and A. Rani, (2014), "Disambiguating Hindi Words Using *N*-Gram Smoothing Models", *International Journal of Engineering Sciences*, Vol.10, Issue June, pp 26-29.

[11] J. Alam, N. Uzzaman and M. khan, (2006), "*N*-gram based Statistical Grammar Checker for Bangla and English", *In Proceedings of International Conference on Computer and Information Technology*.

[12] N. H. Khan, G. C. Saha, B. Sarker and M. H. Rahman, (2014), "Checking the Correctness of Bangla Words using *N*-Gram", *International Journal of Computer Application*, Vol. 89, No. 11.

[13] 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*, vol. 14, issue 1, 2014.

[14] M. R. Rahman, M. T. Habib, M. S. Rahman, S. B. Shuvo and M. S. Uddin, "An Investigative Design Based Statistical Approach for Determining Bangla Sentence Validity," *International Journal of Computer Science and Network Security*, vol. 16, no. 11, pp. 30–37, November 2016.

[15] Q. Qiu et al., "Confabulation based sentence completion for machine reading," *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, Paris, 2011, pp. 1-8.

[16] G. Zweig, C. J. C. Burges. (2011). Tech report: "The Microsoft Research Sentence Completion Challenge".

[17] K. Grabski and T. Scheffer. Sentence completion. *In Proc. SIGIR*, pages 433–439, Sheffield, United Kingdom, 2004.

[18] S. Bickel, P. Haider, and T. Scheffer. Learning to complete sentences. *In Proceedings. ECML*, volume 3720 of Lecture Notes in Computer Science, pages 497{504. Springer, 2005}.

[19] S. Bhatia, D. Majumdar, and P. Mitra. Query suggestions in the absence of query logs. *In Proceedings. SIGIR*, pp. 795-804, Beijing, China, 2011.

[20] M. Shokouhi. 2013. Learning to personalize query auto-completion. *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13)*. ACM, New York, NY, USA.

[21] P.-N. Tan, M. Steinbach, and V. Kumar, "*Introduction to Data Mining*," Addison-Wesley, 2006.

Authors' Profiles

Md. Tarek Habib is pursuing his Ph.D. degree at the Department of Computer Science and Engineering in Jahangirnagar University. He obtained his B.Sc. degree in Computer Science from BRAC University in 2006. Then he got M.S. degree in Computer Science and Engineering (Major in Intelligent Systems Engineering) from North South University in 2009. He is an Assistant Professor at the Department of Computer Science and Engineering in Daffodil International University. His research interest is in Artificial Intelligence, especially Artificial Neural Networks, Machine Learning, Computer Vision and Natural Language Processing.

Abdullah Al-Mamun attained his B.Sc. degree in Computer Science and Engineering from American International University Bangladesh. He received his joint M.Sc. in Computer Science (double degree) jointly from University of Trento, Italy and RWTH Aachen University, Germany under the Erasmus Mundus Double Degree masters program. At present he is working as a Senior Lecturer at the Department of Computer Science and Engineering in Daffodil International University. His research interest includes Data Mining, Artificial Intelligence, Web Search Visualization, Interactive Information Retrieval, Human Computer Interaction, Visual Analytics and Natural Language Processing.

Md. Sadekur Rahman is pursuing his Ph.D. at University Sains Islam Malaysia on Ontology and knowledge management. He obtained his B.Sc. and M.Sc. degree in Applied Mathematics & Informatics from Peoples' Friendship University of Russia. Now he is working as a 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.

Shah Md. Tanvir Siddiquee is a Senior Lecturer in Department of Computer Engineering under the faculty of Science and Information Technology, Daffodil International University, Dhaka, Bangladesh. He holds a B.Sc. in Computer Science and Engineering from Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh and M.Sc. in Computer Science from South Asian University (SAARC

University), New Delhi, India. His research interests in Cloud Computing, Cloud Security using open source cloud environment

Farruk Ahmed is a Professor at the Department of Computer Science and Engineering in Independent University, Bangladesh. He is an eminent professor of Electrical Engineering and Computer Science in Bangladesh. He has had much more than 100 publications in international and national journals and conference proceedings. His areas of research interests in Computer Science are Microprocessor, Microcomputer Based Systems, Natural Language Processing, Speech Processing, Image Processing and Pattern Recognition, Interfacing and Peripherals, Computer Networks, and Information Systems Design.

How to cite this paper: Md. Tarek Habib, Abdullah Al-Mamun, Md. Sadekur Rahman, Shah Md. Tanvir Siddiquee, Farruk Ahmed, "An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction", International Journal of Intelligent Systems and Applications(IJISA), Vol.10, No.2, pp.47-54, 2018. DOI: 10.5815/ijisa.2018.02.05

