Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 2, February 2019, pp. 671~676
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i2.pp671-676

An investigative design of optimum stochastic language model for bangla autocomplete

Md. Iftakher Alam Eyamin, Md. Tarek Habib, Muhammad Ifte Khairul Islam, Md. Sadekur Rahman, Md. Abbas Ali Khan
Daffodil International University, 4/2, Sobhanbag, Mirpur Rd, Bangladesh

Article Info

Article history:

Received Aug 6, 2018
Revised Nov 22, 2018
Accepted Dec 3, 2018

Keywords:

Word prediction
Natural language processing
Language model
N-gram
Machine learning
Eager learning
Performance metric

ABSTRACT

Word completion and word prediction are two important phenomena in typing that have extreme effect on aiding disable people and students while using keyboard or other similar devices. Such autocomplete 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 autocomplete 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.

Corresponding Author:

Md. Iftakher Alam Eyamin,
Daffodil International University ,
4/2, Sobhanbag, Mirpur Rd, Dhaka 1207, Bangladesh.
Email: iftakher.eyamin@gmail.com

1. INTRODUCTION

Innovation in writing and typing of a language is so important. Especially for disable persons and early learners of the language. A person having disability can live a comfortable life if he or she has the opportunity of typing a note, an email or anything else comfortably with the aid of autocomplete. In addition, for the early learners in any field (i.e. students, novice researchers) the autocomplete technique might be beneficial during the learning process by providing most suitable suggestions while searching for new topics with keywords. Though Bangla is one of the most widely spoken languages (3.05% of world population) and considered seventh language of all languages in the world [1], no work was found on automated Autocomplete. In recent couple of years, very few efforts have been made for word prediction, specially focused on Bangla language. In the research work on Bangla word prediction [2], stochastic, i.e. *N*-gram based language models are proposed for completing a sentence by predicting a single word. The next improvement took place in the work of M. T. Habib et. al. [3], where word prediction on sentence by using stochastic, i.e. *N*-gram based language models. They have used a novel metric to assess the performances of their proposed model. Although they achieved good accuracy, it is a matter of fact that opportunities still remain for improvement. Artificial Intelligence used for word prediction in Spanish is also observed in [4], in which using the chart bottom-up technique, syntactic and semantic analysis is done for word prediction. H. Al-Mubaid [5] presented an effective method of word prediction in English using machine learning. In [6] Nagalavi and Hanumanthappa 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 related work use *N*-gram

language model for Autocomplete in Urdu language [7] and in Hindi language [8] for detecting disambiguation in Hindi word. Some research works in Bangla language, e.g. Bangla grammar checker [9] using *N*-gram language model, checking the correctness of Bangla word [10], verification of Bangla sentence structure [11], and validity determination of Bangla sentences [12] are also conducted. There are some different word prediction tools such as AutoComplete by Microsoft, AutoFill by Google Chrome, TypingAid, LetMeType etc. In [13] 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. An *N*-gram model is constructed in [14], which was used to compute 30 alternative words for a given low frequency word in a sentence, and human judges then picked the best impostor words, based on a set of provided guidelines Index-based retrieval algorithm and a cluster-based approach are proposed at [15] for sentence-completion. Bickel et al. [16] learned a linearly cast *N*-gram model for sentence completion. Bhatia et al. [17] extracted frequently occurring phrases and *N*-grams from text collections and deployed them for generating and ranking auto-completion candidates for partial queries in the absence of search logs. A new approach is proposed in [18], for learning to personalize auto-completion rankings based.

Word prediction means guessing the next word in a sentence. Auto complete or Autocomplete works so that the user types the first letter or letters of a word and the program provides one or more higher probable words. If the word he intends to type is included in the list, he can select it, for example by using the number of keys. If the word that the user wants is not predicted, the user must type the next letter of the predicted word. At this time, the word choice(s) is altered so that the words provided begin with the same letters as those that have been selected or the word that the user wants appears it is selected.

Autocomplete technique complete word by analyzing previous word flow and first letter of the word for auto completing a word and sentence with more accuracy and reduces misspelling. *N*-gram language model is important technique for word prediction.

The problem addressed in this paper is about stochastically predicting a suitable word to complete an incomplete sentence, which consists of some words and a single character. Let $w_1w_2\ w_3\ \ldots\ w_{m-1}\ w_m$ be a sentence i.e. sequence of words, where $w_m = c_1c_2c_3\ldots\ldots c_n$ and $w_1w_2\ w_3\ \ldots\ w_{m-1}c_1$ has already been typed. The problem is to build a language model which takes $w_1w_2\ w_3\ \ldots\ w_{m-1}c_1$ as input and predicts an *n*-tuple of word fragments (v_{m1},v_{m2}, v_{m3}, $\ldots v_{mn}$) as output in order to match the remaining untyped word fragment $c_2\ c_3c_4\ldots\ldots c_n$, as shown in Figure 1.

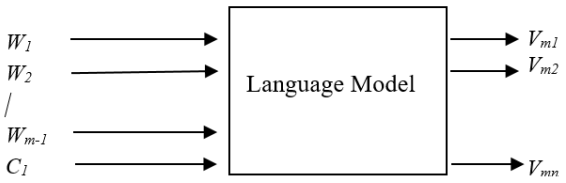

Figure 1. Language Model

We use large data corpus for training in *N*-gram language model for completing correct Bangla word to complete a Bangla sentence with more accuracy. In this paper, we propose an investigating design of optimum stochastic language model for Bangla autocomplete using supervised machine learning technique based on different *N*-gram language modeling. Probability is based on counting things or word in most cases. In our previous works [2, 3], we used different types of language models for word prediction. Both these two works are different from the work presented in this paper because word prediction obviously differs from autocomplete, i.e. word completion. In these earlier works word [2] or word set [3] is predicted based on one or more preceding words, but in this work, word fragment set is being predicted based on one or more words and a single character.

The rest of the paper is organized as follows. In Section 2, comes the description of our approach to solve the problem. Section 3 describes how we apply our entire methodology and what results are achieved. In Section 4, 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 5.

2. PROPOSED METHOD

We begin with five language models, namely unigram, bigram, trigram, backoff and linear interpolation. All these language models are based on *N*-gram approximation. Bayesian classifiers have been

used in [19-21]. As opposed to Bayesian, classifier assumes no correlation between words in the same text, where *N*-gram language model assume relationships between the words, and evaluate the probability of a word being before or after another word. The ordinary equation for the *N*-gram approximation to the conditional probability of the next word in a sequence is:

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n-1}) \quad (1)$$

Equation 1 shows that probability of a word w_n given all the previous words can be presumptive by the probability given only the previous *N* words. If $N = 1, 2, 3$ in (1), the model becomes unigram, bigram and trigram language model, respectively, and so on.

We train our full corpus along with our five language models. We calculate each word *N*-gram portability. We predict the word, which have the most frequency based on *N*-gram probability. Then we check the first character of the predicted word with our given character. If the first character of the predicted word and given word matched we select the word as predicted or result word.

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. If is mention that the equation 2 & 3 used from our previous paper [3]. If w_m matches with v_{mi}, (i.e. w_m equals v_{mi}, where $1 \leq i \leq n + 1$), then the accuracy is

$$Accuracy = \frac{n+1-i}{n} \times 100\% \quad (2)$$

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

$$Failure\ rate = \frac{f}{p} \times 100\% \quad (3)$$

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.

3. RESEARCH METHOD

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, $|w|$. 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 of [3]. The best model is chosen by the technique, which was describe in our previous paper [3].

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 and a character 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 12 million (12,203,790) words and about 1 million (937,349) sentences, where total number of unique words is 294,371 and average word length ($|w|$) is 7.

During this work, we divided the entire corpus into two parts, namely training part and testing part. The holdout method [22] 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 seven (7) 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 1of [3].The entire approach is shown in Figure 2.

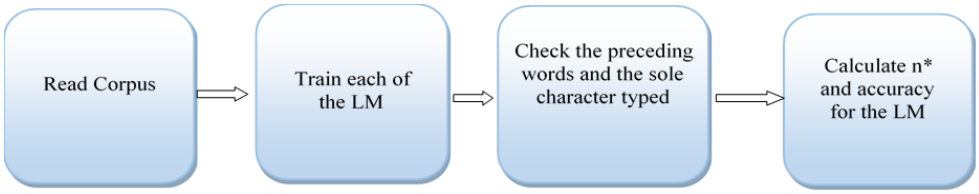

Figure 2. Proposed approach for specific model

4. RESULT AND DISCURTION

The optimum prediction length (*n**) along with the accuracy of each model is shown on Table 1.

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	3.4%	12.65%	21.32%	25.37%	29.48%	33.75%	38.62%	7
Bigram	59.90%	65.35%	69.14%	72.52%	74.95%	77.06%	79.12%	7
Trigram	75.74%	79.21%	81.02%	81.93%	82.48%	83.03%	83.38%	7
Backoff	75.74%	80%	82.39%	83.41%	84.73%	85.42%	85.96%	7
Linear Interpolation	73.76%	79.70%	83.33%	86.39%	88.42%	90.10%	91.58%	7

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. After finding out the different accuracy rate of top three models 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 Figure 3) 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. After finding out the different failure rate of top three models with the test set consists of sentences with different lengths, the average failure rate (see Figure 4) 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.

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

Available Words in Test Sentences	Accuracy of Language Model		
	Trigram	Backoff	Linear Interpolation
1	15%	20%	37.78%
2	34.22%	44%	61.07%
3	59.99%	74.16%	63.57%
4	60.68%	76.21%	64.56%
5	61.30%	79.2%	73.21%
6	61.52%	79.99%	73.40%
7	67.5%	82.63%	74.28%
8	68.26%	83.43%	74.34%
9	68.69%	85.40%	75%
10	70.12%	85.75%	75%
11	70.71%	87.40%	76.73%
12	71.59%	90%	82.63%
13	83.57%	91.11%	83.92%
14	83.57%	91.99%	92.14%
15	87.85%	93.6%	93.57%

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

Available Words in Test Sentences	Failure Rate of Language Model		
	Trigram	Backoff	Linear Interpolation
1	64.44%	54.28%	60.00%
2	39.28%	25%	35.71%
3	38.63%	21.62%	30.43%
4	36.95%	20%	28.57%
5	36.95%	19.04%	25%
6	32.14%	16.34%	25%
7	30.43%	15.78%	25%
8	30.43%	13.51%	23.91%
9	28.76%	12.12%	22.72%
10	28.76%	12.04%	21.73%
11	28.57%	10.71%	19.56%
12	27.27%	9.52%	14.54%
13	14.28%	7.40%	14.28%
14	14.28%	5.71%	7.14%
15	10.71%	4.21%	3.57%

During experiment, as shown on Table 1, it is noticeable, the top three models have shown good accuracy among all the models, 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 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. Moreover, from the average accuracy and average failure rate of all models (Figure 3 and Figure 4 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 Figure 5.

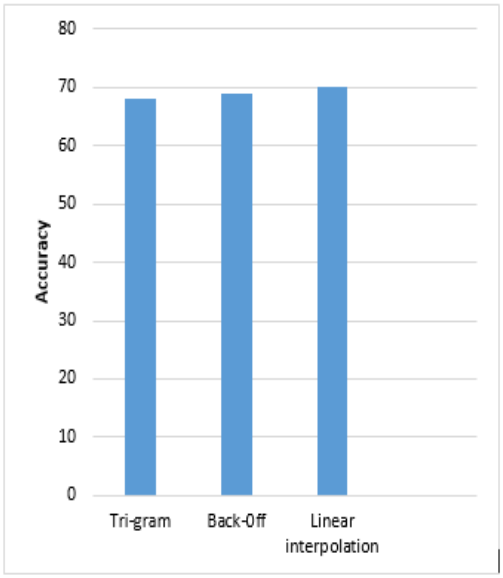

Figure 3. Average accuracy of language models

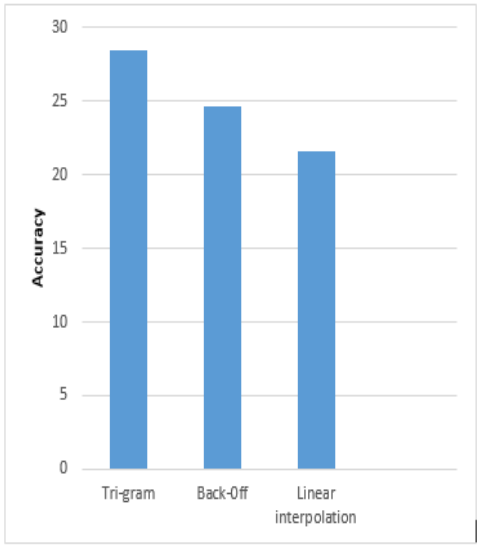

Figure 4. Average failure rate of language models

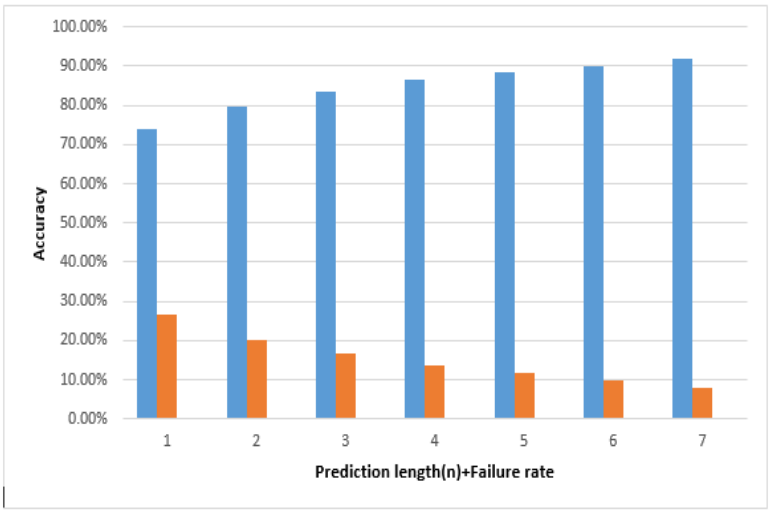

Figure 5. Accuracy and Failure along with the prediction length of Linear Interpolation model

Although the linear interpolation model has shown better performance than other top models (91.98% with n* = 7) and the experiment result is promising.

5. CONCLUSION

The focus of this research was modeling, training and apply techniques that can assist in automatic Bangla word completion. For the purpose of this research, a large and rich Bangla corpus is applied and supervised machine learning technique based on popular *N*-gram language model is used. Among five-language model to determine the best language model is the main contribution of the research. Though during the several phases of experiments, 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. An adaptive software for Bangla automated word completion based on this work will be developed.

REFERENCES

[1] List of languages by number of native speakers, Available at: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers. (Last Accessed: March 10, 2018).
[2] 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*. November 2015, vol. 5, no. 6, pp. 67–75.
[3]. M. T. Habib, A. Al-Mamun, M. S. Rahman, S. M. T. Siddiquee and F. 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)*, February 2018, vol. 10, no. 2, pp. 47-54.
[4] 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.
[5] H. Al-Mubaid, "A Learning-Classification Based Approach for Word Prediction". *The International Arab Journal of Information Technology*, 2007,Vol. 4, No. 3.
[6] 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.
[7] Q. Abbas, (2014). "A Stochastic Prediction Interface for Urdu". *Intelligent Systems and Applications*, Vol.7, No.1, pp 94-100.
[8] U. P. Singh, V. Goyal and A. Rani. "Disambiguating Hindi Words Using *N*-Gram Smoothing Models". *International Journal of Engineering Sciences*.2014, Vol.10, Issue June, pp 26-29.
[9] J. Alam, N. Uzzaman and M. khan. "*N*-gram based Statistical Grammar Checker for Bangla and English". *In Proceedings of International Conference on Computer and Information Technology*. 2006.
[10] 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*, 2014, Vol. 89, No. 11.
[11] 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*. 2007, vol. 14, issue 1.
[12] 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*. November 2016, vol. 16, no. 11, pp. 30–37.
[13] 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.
[14] G. Zweig, C. J. C. Burges. Tech report: "The Microsoft Research Sentence Completion Challenge". 2011.
[15] K. Grabski and T. Scheffer. Sentence completion. *In Pro c. SIGIR*, pages 433–439, Sheffield, United Kingdom, 2004.
[16] S. Bickel, P. Haider, and T. Scheffer. Learning to com-plete sentences. *In Proceedings. ECML*, volume 3720 of Lecture Notes in Computer Science, pages 497{504. Springer, 2005}.
[17] S. Bhatia, D. Majumdar, and P. Mitra. Query suggestions in the absence of query logs. *In Proceedings. SIGIR*. Beijing, China, 2011, pp. 795-804.
[18]. Daniel Jurafsky and James H. Martin. Speech and Language processing, USA: Prentice-Hall, Inc. 2000.
[19] K. C. Rani, Y. Prasanth. "A Decision System for Predicting Diabetes using Neural Networks". *IAES International Journal of Artificial Intelligence (IJ-AI)*. June 2017, Vol. 6, No. 2, pp 56-65.
[20] S. Shah, K. Kumar, Ra. K. Saravanaguru, "Sentimental Analysis of Twitter Data Using Classifier Algorithms". *International Journal of Electrical and Computer Engineering (IJECE)*. February 2016, Vol. 6, No. 1, pp. 357-366.
[21] Ashwin V, "Twitter Tweet Classifier", "*IAES International Journal of Artificial Intelligence (IJ-AI)*". March 2016, Vol. 5, No. 1, pp. 41-44.
[22] P.-N. Tan, M. Steinbach, and V. Kumar, "*Introduction to Data Mining*," Addison-Wesley, 2006.