Prithvi Academic Journal
[A Peer-Reviewed, Open Access Multidisciplinary Journal]
Indexed in NepJOL with JPPS Star Ranking
ISSN 2631-200X (Print); ISSN 2631-2352 (Online)
URL: https://ejournals.pncampus.edu.np/ejournals/paj/

Centre for Research & Innovation
Prithvi Narayan Campus
Tribhuvan University
Pokhara, Nepal
http://pncampus.edu.np/

ORIGINAL RESEARCH ARTICLE

The Use of N-Gram Language Model in Predicting Nepali Words

Bal Ram Khadka, Er.
B.Sc. CSIT Program, Prithvi Narayan Campus, Pokhara, Nepal

Article History: *Submitted 30 January 2022; Reviewed 29 March 2022; Accepted 6 April 2022*
Corresponding Author: *Bal Ram Khadka*, ***Email***: *balramkhadka@pncampus.edu.np*
DOI: *https://doi.org/10.3126/paj.v5i1.45040*

ScanToAccess eCopy

ABSTRACT

This paper aims to study the problems of automated generation and understanding of natural human languages. The word prediction and word completion from a tab-complete in typing is particularly useful to minimized keystrokes for the users with specific necessaries, and to reduce mistakes, and typographic errors. The word prediction techniques are well-established methods that are frequently used as communication aids for people with disabilities to accelerate the writing, to reduce the effort needed to type and to suggest the correct words. It is something that is skillful at doing prediction according to the previous context. Projection can either be established on word figures or verbal rules. The N-gram model is about predicting n^{th} word from N-1 words. It assigns the probabilities to sentences and sequences of words of all possible combination of n words. To meet the objective, this research uses statistics amount of Nepali language of diverse word kinds to expect right word with as much precision as possible. Under the statistical method, this research will deal with the N-gram method to predict the next word for the Nepali language using Viterbi as decoding algorithm.
KEYWORDS: Word suggesting, Katz model, Viterbi model, N-gram, back-off smoothing

INTRODUCTION

With the advent of internet, the use of web-based information helps us to spread in generating the texts. The web and other request extract value from Twitter and Facebook comments, the application as click-based, customer segmentations and sentiment analysis used that information and get the value of text. In this process, there is a problem of examining a large corpus of words and uncovering the structure and arranging the text in order to detect the corpus from computational methods. The heart of this research is to take a corpus from various bases, to clean and to examine that corpus and to define a predictive model for offering the more comprehensive text, based on the past two or more texts provided by a consumer. The client input could reach from formal, specialized communication styles to informal, short messages. So, the knowledge of its characteristics to detain the corpus is critical. For example, a client gives the word into the company's device such as "प्रदेशको राजधानी". A prediction model would list the most likely word, for what the next text might be such as "पोखरा," "तोकने," or "दुईतिहाइ" (Khadka, 2020).

Data sciences are gradually producing the use of natural language processing merged with arithmetic approaches to describe and influence the flows of data that are text founded and not natural quantities. There are many methods available in the Python language to programing quantitatively in text-based information. The major focus of this study is to identify a technique that helps clarify accuracy and scalability for large data sets. This research makes all the code and algorithms available as an open and collaborative code for others to investigate and improve. The objectives of this research are:

- To predict next word for sentence completion using the N-gram statistical method with back-off smoothing.
- To analyze and compare the performance of back-off smoothed trigram (3) and quad gram (4) language model using Viterbi algorithm.

The significance of this paper is to increase the pace while entering the words into the computer and to reduce missing and suggesting words in the Nepali language, by offering word projection and auto accomplishment. So it produces a helpful situation for the users that are available in the Nepali language. Additionally, the findings of this study are important for public to assist in the Nepali language. The computer professionals who want to custom the Nepali language in their design will also learn this technique from this paper.

In addition, this paper provides an advanced knowledge of the Nepali language by constructing accessibility, ease of utilization and reliability for scientific and scholarly groups. It also provides skills of working on Nepali language, using the Natural Language Processing (NLP), particularly in error, sense disambiguation, machine conversion and text handling so that individuals working in the area can make logic disambiguation and auto-accomplishment of the Nepali language.

LITERATURE REVIEW

The prediction of user behavior assists systems to address the problem of envisaging the repetitive tasks that can accomplish easily and can get maximum throughput. Bickel et al. (2005) advanced a sentence achievement method established on N-gram language models. It has the greatest precision and recalls performance metrics. This research shows N-gram based achievement that has enhanced more precision recall sketch than index based retrieval. In addition, the sentence completion is effective that sharply depends on the document collection.

Agrawal et al. (2011) have worked on preprocessing, graph scooping, and hashing for making the hint list. The hint list provided by our advised style is more appropriate because the list is a manuscript aspect. Test outcomes after this offered style show the significantly good performance, using the document specific auto-complete search techniques.

It delivers ideas to the client by getting immediate texts from the graph, which is shaped from the keyword. For performance evaluation, it has performed experiments on the tool using the proposed method. The results showed that the proposed approach significantly outperforms the available methods based on history or commonly defined words and shrinkage exploration period expressively. The projected approach was liked by the applicants. The application used for investigation adopted two permits of subject to be examined to form a graph.

K.C.'s (2012) study reported that a sentence prediction based on N-gram model with Viterbi algorithm as decoding algorithm is a dynamic algorithm. By using the measuring criteria Precision, Recall and F-Score, the list of bigram and trigram token specifies the prediction of the sentence based on the probability of the N-gram. The test results from this proposed method show 60.4 % and 45.6% for trigram and bigram language models respectively. The study provides suggestion by increasing the corpus size and of N-gram, the accuracy of prediction of the system will increase.

Rani et al.'s (2014) study is based on the backoff and delete interpolation comparison. The accuracy of word sense dis-ambuigator depends upon a number of entries in the database and size of N-gram used. The test results from this proposed method show 65 accuracy for the backoff model and 67 accuracy for delete interpolation where both corpuses are tested on a set of 10 highly polysemous Hindi words. It provides a suggestion: database can be extended to include more entries to improve the accuracy.

Haque et al.'s (2015) study is based on the comparison of trigram and backoff smoothing language models. It begins with a training corpus of 0.25 million words size. The corpus has been formed from the Bangla newspaper. The corpus contains 14,872 words forms. The test results from this proposed method show 63.04 accuracy for trigram model, 63.50 accuracy for backoff model. It provides suggestion to the user by rise data corpus size to get higher implementation.

With the advent of internet, the text-based information value is also increased. So to analyze a corpus and figure out the structure of the words presented in corpus is most valuable or a key concept in recent trends. Gendron et al. (2015) used the Blog, News, Tweets and Corpus in four languages (English, German, Finnish and Russian). It used R text-mining package 'tm'. It used Good-Turing approach to develop tables of count. In addition, the Katz backoff approach was implemented if a word was not found in the table. This analysis suggests that the predictive model can be built, but data product is most useful for prediction. The refinement of data product allows more robust seen, making use of input.

Predicting the most probable word enhances the communication experience. Due to fast paced nature of conversion, the text prediction application increases the rate of communication. Dumbali et al. (2019) used Good Turing algorithm for smoothing the training data. The data needed for training and testing of development are derived from blogs, tweets and news and the combined size of the data is about 556 MB. It cuts the N-gram with total frequency below 3. This paper shows that the accuracy of the model can be seen as improving when the values of N-grams increases.

The text prediction aims to prevent the period and effort by easing the keystrokes and develop text excellence by misspelling avoidance. Yazani et al.'s (2019) study is based on the Trigram language model. It measures the declining moment, keystrokes and text making. This method reduced keystrokes to type narrative texts, the typing interval time of free text that falls by 33.36% on average as compared with the similar research. The text generation rate for this system is 0.61 word per second.

This paper shows that the systems accelerate the free text entry, using the statistical word prediction.

METHODOLOGY

The word estimation model identifies a number of words present in the context while the previously appeared words in the context determine the next word in the prediction. So the next word may depend on the previous word form in the sentence.

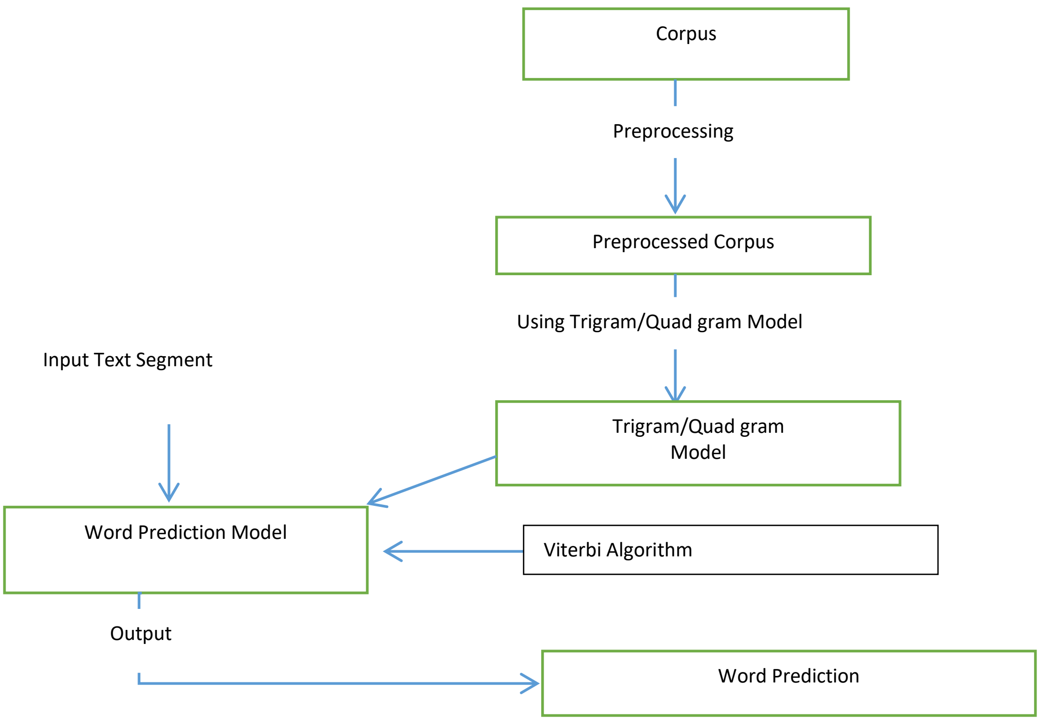
In order to perform the word estimate model for the Nepali language, various steps are used in this paper:

- Texts or corpus pre-processing, which receive the corpus, then tokenize that word, eliminate the stopping word or end word and implement normalization. To the pre-process corpus, the tool I used is Microsoft Visual Studio 2010 Professional Edition. After raw bigram probability is identified, the N-gram are stored in Microsoft SQL Server 2005 for intermediate result stored.
- The N-gram word provides the frequency based on the rate of co-occurrences of the text prediction that is done. In order to perform the prediction of next word, the users should give the token of word and the most probable word is predicted by the model.

The structure of the word prediction is presented in Figure 1.

Figure 1

Framework of Word Prediction



Corpus

This section is arranged through the gathering of data and preprocessing of the gathered data. For this report, I collected the corpus from मदन पुरस्कार पुस्तकालय, which is transferred to Unicode and XCES format (Khadka, 2020). The corpus is used for training and testing of language model.

Corpus Pre-processing

The corpus pre-processing helps to build the fresh data correct for examination by refining the data condition. The preprocessing step makes input ready for the language model. In preprocessing step,

- Firstly, the file containing each word in the corpus in the new line.
- Secondly, insertion of <s> and </s> in the corpus to define the line of sentence.
- Finally, each sentence in the corpus ends with the end symbol ('।'), joins the symbol ('-'). Then, other terminal symbols are removed and the sentence is split into different bigram, trigram and quad gram token.

Implementation Tools

Microsoft Visual Studio 2010 Professional

In this experiment, the algorithm was implemented in the Microsoft Visual Studio 2010 Professional Edition which run on the corpus. The Microsoft Visual Studio 2010 Professional is wonderful, extensive and expensive from one outlook. It is an organized area that develops important tasks of making, troubleshooting and delivering applications. If we give Visual Studio 2010 Professional a chance to release the original capacity, it successfully transmits your thoughts, Easy ins, Executive suite and Easy on the eyes (*Microsoft-visual-studio-2010-professional-free-download*, n.d.).

Microsoft SQL Server 2005 Express Edition

To view the result and store the N-gram result, the Microsoft SQL Server 2005 Express Edition is used. It is a prevailing and consistent data management artifact that brings fine descriptions, data safety and operation for embedded remedy clients, light web applications, and local data stocks. Intended for easy deployment and quick prototyping, it is free at no cost and you are free to restructure it with products. It is aimed to integrate effortlessly with your other server setup investments (*Microsoft SQL Server 2005 Express Edition*, n.d.).

Algorithm Approaches

In this research, the model is trained using pre-processed corpus by calculating raw bigram probability. The pre-process corpus is divided into five folds and each fold is divided into four parts as training and a part as testing token. The description of this algorithm is as follows:

Trigram/Quad Gram Language Model using Backoff Smoothing

The training data are generated by the trigram and quad gram language model. The training data consist of the text from the pre-processing corpus and testing data for the model that is obtained from the original corpus (Khadka, 2020).

The major problem with the maximum likelihood estimation is training the factors of an N-gram representation. The drawback of sparse data produced by the fact that maximum likelihood estimate was built on a specific set of training data. Any N-gram that happened enough quantity of periods has a reliable approximation of its possibility. But because any corpus is inadequate, some completely adequate English word arrangements are connected to be lacking from it. These lost data channels that the N-gram matrix for any particular training corpus are confident to have a very copious digit of cases of assumed – zero possibility N-grams that should truly have some non-zero probability. Furthermore, the MLE technique also creates low guesses when the total is non-zero, however, quite small.

Smoothing methods usually block zero probabilities, but they also endeavor to progress the accuracy of the model as a whole. Whenever a probability is calculated from little counts, smoothing has the possibility to knowingly advance calculation.

One way to resolve this difficult is called Katz back-off. Katz back-off is a propagative n-gram language classical that guesses the qualified possibility of a text allowing its past in the N-gram. It achieves this valuation by backing off through progressively shorter past copy in some situations. By doing so, the model with the highly dependable information, roughly a known past, is used to deliver the improved results.

This type was presented in 1987 by Slava M. Katz., N-gram language modes were built by training separable shows for another N-gram regularities using maximum likelihood estimation and then adding them concurrently (*Katz's back-off model*, n.d.).

$$P_{katz}(W_{i-3}W_{i-2}W_{i-1}) = \begin{cases} P^*\left(\frac{W_i}{W_{i-3}W_{i-2}W_{i-1}}\right), & if\ C(W_{i-3}W_{i-2}W_{i-1}W_i) > 0 \\ \alpha(W_{i-3}W_{i-2}W_{i-1})P^*\left(\frac{W_i}{W_{i-2}W_{i-1}}\right), & else\ if\ C(W_{i-2}W_{i-1}W_i) > 0 \\ \alpha(W_{i-2}W_{i-1})P^*(W_i/W_{i-1}), & else\ if\ C(W_{i-1}W_i) > 0 \\ (W_i)P^*(W_i), & otherwise \end{cases} \quad (1)$$

Here P^* is defined as the discounted assessment of the conditional possibility of an N-gram.

C(X) = no. of instances X performs in training

W_i = i_{th} text in the particular environment

Viterbi Algorithm

Viterbi algorithm is a dynamic programming algorithm to discover the best likely order of unknown states called Viterbi path, that result is an order of viewed events, especially in the circumstance of Markov information resources and hidden Markov Models (K.C., 2012). The problem is to find the most likely word sequence $W_{t+1} \dots\dots W_{t+T}$ given an initial word sequence $W_1 \dots\dots W_t$.

$$\underset{W_{t+1}\dots W_{t+T}}{\operatorname{argmax}} P\left(\frac{W_{t+1}\dots\dots W_{t+T}}{W_1 \dots\dots W_t}\right) \quad (2)$$

Let a_{ij} be the transition probabilities from the state i to j, π_i the initial possibility of being in the state i. S is the set of possible states.

$$V_{1,k} = p\left(\frac{O_i}{X_1^k}\right)\pi_k \forall_k \in S$$

For each time slice k $\in S$

$$V_{t,k} = \max_{i \in S} p(O_t / X_t^k)\, a^k V_{t-1,i} \quad (3)$$

$$bp_{t,k} = \underset{i \in S}{\operatorname{argmax}}\, p(O_t / X_t^k)\, a^{ik} V_{t-1,i} \quad (4)$$

Thus, V_{tk} contains the maximum possibility of presences in the state k at time t and bp_{tk} contains the state that had the maximum probability

Maximum Likelihood Estimation

The maximum likelihood estimation (MLE) is a technique of approximating the factors of an expected probability supply, giving some practical data. This is realized by enlarging a likelihood function so that, under the implicit arithmetic model, the practical data is most likely. The logic of maximum likelihood is both natural and adaptable.

K Fold Cross Validation

This system comprises randomly splitting the dataset into k groups or folds of roughly uniform size. The first fold is kept for testing and the model is trained on k-1 folds.

The activity is iterated K times and each time unique fold or a unique group of data points are used for validation.

Proposed Algorithm

Step1: Divide the pre-processed corpus into k-fold (k1,..k5)
Step 2: Identify the unigram, bigram, trigram, quad gram for each fold
Step 3: Calculate the raw bigram, trigram and quad gram probability in each fold
Step 4: a) given user input
b) Calculate the prediction of next word by using Viterbi Algorithm (MLE)
c) Find the maximum probability of word, which is the predicted words
Step 5: Stop Prediction

TESTING AND ANALYSIS

Corpus Data Statistics

The different corpus is merging together and contains 1596 lines of Unicode Nepali sentence, and store in the SQL database with table name tblsentence1. The merge corpus is divided into five folds; each fold is dividing into training and testing set with ratio 80:20. The N-gram words and prediction words are stored in SQL database with table name tblngrammodel and tblprediction respectively and the structure of fold is:

Table 1

Structure of Word in the Model

N-gram		K-fold-1	K-fold-2	K-fold-3	K-fold-4	K-fold5
Unigram	Training	6206	5881	6244	6235	6329
	Testing	2375	2886	2309	1994	1759
Bigram	Training	21747	20510	22194	22534	23111
	Testing	6607	8024	6116	5345	4572
Trigram	Training	27998	26515	28840	29503	30692
	Testing	8176	9679	7283	6416	5171
Four gram	Training	28521	27112	29558	30416	31813
	Testing	8419	9817	7353	6424	5031

Training and Testing Data

The training and testing data is prepared from the original corpus. They consist of the word in table name tblngrammodel and tblprediction (Khadka, 2020) where the training data field identifies the sets that is training or testing. The training is performed by using raw bigram probability calculation. The raw bigram probability is calculated by using the frequency of past word and the frequency of the predicted word in the table,

$$raw\ bigram\ probability\ of\ words = \frac{frequncy\ of\ predicted\ gram\ word}{frequcny\ of\ the\ past\ word} \quad (1)$$

Table 2

Structure of Unigram Word in Testing

Unigrams Word	Frequency
को	678
मा	377
ले	299
का	234
हरू	186
लाई	132
भए	102
नेपाल	67
बाट	62
पनि	56
Top 10 Words	2193
Remaining	6783

Figure 2

Pie Chart of Unigram Testing Word in Kfold1

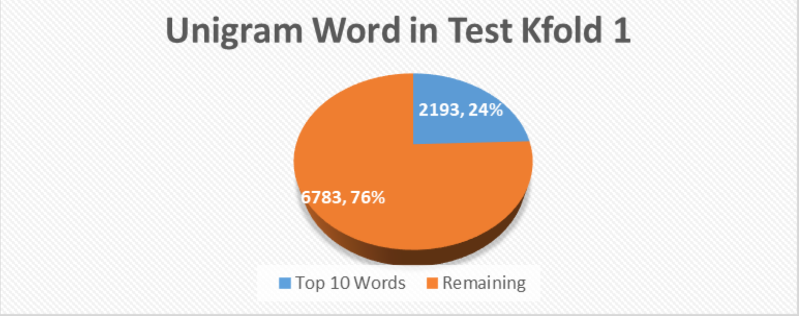


Figure 3

Pie Chart of Top 10 Unigram Testing Word in Kfold1

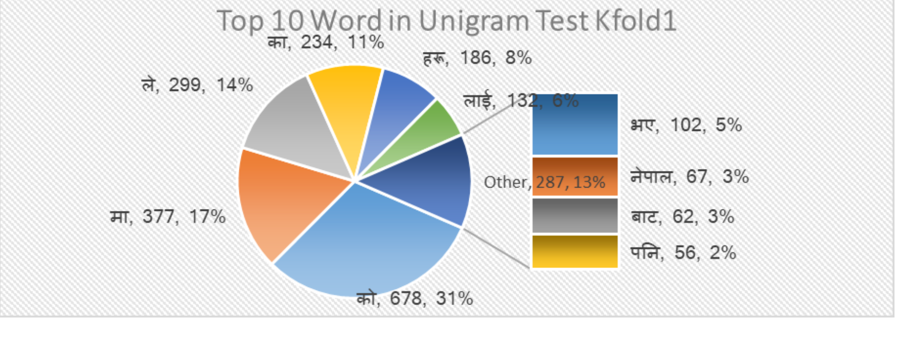


Table 3
Structure of Unigram Word in Training Kfold1

Unigrams Word	Frequency
को	2189
मा	1257
का	844
ले	815
हरू	706
लाई	513
पनि	276
भए	249
बाट	212
छन्	158
Top 10 Words	7219
Remaining	23091

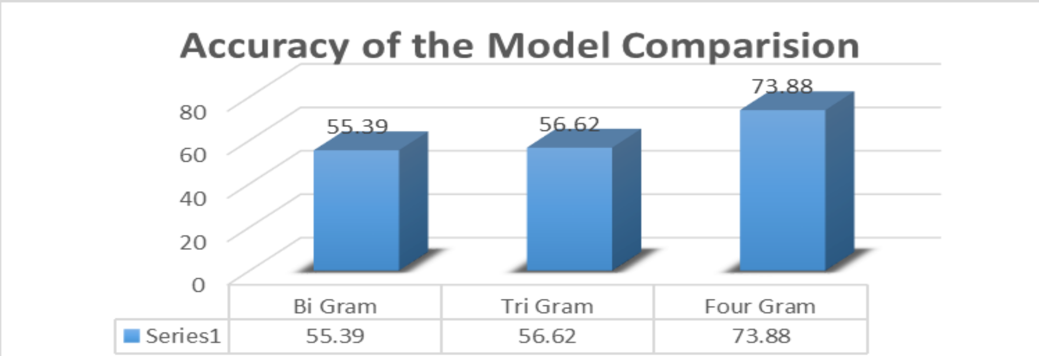
Table 4
Input and Output of the Prediction Model

S.N.	Test N gram Word	Input Word	Predicted Word	Kfold	Ngram	Prediction Status
1	स्थानीय निकाय	स्थानीय	कांग्रेसी	1	Bigram	0
2	निकाय हरू	निकाय	को	1	Bigram	0
3	हरू को	हरू	को	1	Bigram	1
4	मैदान मा	मैदान	मा	2	Bigram	1
5	माउत्रे	मा	पनि	2	Bigram	0
6	उत्रे का	उत्रे	का	2	Bigram	1
7	कोइराला ले	कोइराला	ले	3	Bigram	1
8	ले भन्नुभयो	ले	आफ्नो	3	Bigram	0
9	सरकार ले	सरकार	ले	3	Bigram	1
10	हरू बाट	हरू	को	4	Bigram	0.1
11	हरू ले	हरू	को	4	Bigram	0.9

12	इनपुट फर्म	इनपुट	फर्म	5	Bigram	1
13	फर्म डिजाइन	फर्म	हरू	5	Bigram	0.5
14	हरू को प्रशिक्षण	हरू को	संख्या	1	Trigram	0
15	गरिए को थियो	गरिए को	थियो	1	Trigram	1
16	भए को ले	भए को	थियो	3	Trigram	0.2

Figure 4

Bar Diagram of Prediction Model Bigram, Trigram and Four-gram Accuracy



Analysis of the Prediction Model

Given an initial word, a model predicts next word of the Ngram, the purpose of the model is to write or predict the most likelihood text from the suggested word. This suggestion is based on the Viterbi likelihood concept where the maximum probability of the word is considered or taken as a predicted word.

For the evaluation of accuracy, this research calculates accuracy in each fold and the entire model. The accuracy is determined by number of time the accurate prediction and the total number of time the prediction done by the model Bickel et al. (2005).

$$Accuracy = \frac{\text{total number of word predicted correctly}}{\text{total number of prediction}} \qquad (4.2)$$

Table 5

Prediction of Bigram Word in Model

S.N.	Kfold	Correct Prediction	Total Prediction	Accuracy Percentage
1	Kfold 1	2.6	8	32.5
2	Kfold 2	6.7	8	83.75
3	Kfold 3	5	10	50
4	Kfold 4	2.5	7	35.71
5	Kfold 5	6	8	75

Accuracy of Bigram Model is = 55.39

Table 6

Prediction of Trigram Word in Model

S.N.	Kfold	Correct Prediction	Total Prediction	Accuracy Percentage
1	Kfold 1	2	4	50
2	Kfold 2	7.4	9	82.22
3	Kfold 3	6.2	9	63.33
4	Kfold 4	2	4	50
5	Kfold 5	1.6	5	32

Accuracy of Trigram Model is = 56.62

Table 7

Prediction of Four-Gram Word in Model

S.N.	Kfold	Correct Prediction	Total Prediction	Accuracy Percentage
1	Kfold 1	3	4	75
2	Kfold 2	2.5	3	83.33
3	Kfold 3	3.8	6	68.88
4	Kfold 4	No prediction		
5	Kfold 5	No prediction		

Accuracy of Four-Gram Model = 73.88

DISCUSSION OF THE RESULTS

The result is discovered in increasing accuracy. By analyzing the result, Four-gram Prediction Model generates 73.88% accuracy and Trigram Prediction Model generates 56.62 % whereas Bigram Prediction Model generates 55.39% accuracy (Khadka, 2020). From this finding, it is clear that Four-gram and Trigram Prediction Model are more accurate than Bigram Prediction Model. In this paper, five k-fold is divided into training and test 80:20 ratio each where test subsets are not used in training phase to avoid the biased result. Some folds show the higher rate of prediction and some are not. This happens due to the word is present in the higher prediction fold that is repeated and less prediction fold that is the unique or few repletion of word. If the word is repeated maximum, then the prediction is increased in percentage. With this goal, to increase accuracy and to reduce data footprint, I used to limit the N-gram applying two criteria:

- Cutting the N-gram with total frequency below 3.
- N-gram words frequency below 3 are not considered in prediction because the use of the prediction accuracy is decreased and the below 3 is considered as rear. So the rear element of the text can't be more effective in prediction.
- To recognize the N-gram prediction,
  - ✓ Selecting the word in the top by weight 1 if it is the correct prediction
  - ✓ If the prediction is in the list, the use of prediction percentage and the weight are calculated with the proportionating method.
- If the prediction is not in the list, then the weight 0 is assigned.

CONCLUSION

This research has successfully predicted the word with the input word fragments, using Bigram, Trigram and Four-gram Language Models with Viterbi algorithm as a decoding algorithm. By using the measuring accuracy, this paper found out that the Four-gram Language Model predicts the word more accurately than the Tri gram language model and the Bigram language model. This research also identifies that when the number of N-grams is increased, the number of words are repeated or next word is the same. This is why the longer in N-gram, the more accuracy is increased due to the repeated words. This research originates 73.88% accuracy for Four-gram and 56.62% for the Trigram language model whereas 55.39% for the Bigram language model (Khadka, 2020). This research used the corpus, which consisted of 1596 lines of sentences and 39,286 Unicode Nepali words ~~is~~ are used for training and testing. Due to the limitation of the size of the words in the corpus, the number of bigram, trigram and four gram counts are moderate. So, when this research predicts the word segment based on the Bigram, Tri gram and Four-gram counts, the results are in moderate accuracy. So to increase the accuracy, the corpus size should be increased. By increasing the corpus size, the results are in better position than this study has found out.

RECOMMENDATIONS

This research has predicted the words using Bigram, Trigram and Four-gram Language Models. Further, 5-gram, 6-gram, up to N-gram can be used to predict the words. By increasing the rate of N, the precision of the system may increase. Not only the word completion, but also the grammar judgment, word sense disambiguation and steaming of word in the sentence task can be performed. In addition, the size of corpus could be increased to get the better result. Furthermore, the machine learning method and other methods can be used to complete the word prediction.

ACKNOWLEDGMENTS

This research is based on my Master Thesis entitled "Predicting Word using N-Gram Language Model for Nepali Language," submitted to Nepal College of Information Technology, Pokhara University, Kathmandu I would like to extend my sincere gratitude to my respected supervisor Dr. Bal Krishna Bal, Kathmandu University for his constructive criticism and intellectual support. I would also like to express my gratitude to the respected teachers Mr. Saroj Shakya, Program Co-ordinator, NCIT, Nepal for granting me this opportunity to work on this research.

REFERENCES

Agrawal, N. & Swain, M. (2011). Auto complete using graph mining: A different approach. https://doi.org/10.1109/SECON.2011.5752947

Bickel, S., Haider, P. & Scheffer, T. (2005). Predicting Sentences using N-Gram language models. *Proceeding of Conference on Empirical Methods in Natural Language Processing*, HLT'05, 193–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dumbali, J., & Rao A., N. (2019). Real time word prediction using N-grams model. *International Journal of Innovation Technology and Exploring Engineering (IJITEE)*, *8*(5), 870–873.

Gendron, Jr., G. R. G. J., Consulting, C., & VA, C. (2015). Natural language processing: A model to predict a sequence of words. *MODSIM World*, *13*, 1–10.

Haque, Md & Habib, Md & Rahman, Md. Mokhlesur. (2015). Automated word prediction in Bangla language using Stochastic language models. *International*

Journal in Foundations of Computer Science & Technology, *5*, 67-75. https://doi.org/10.5121/ijfcst.2015.5607

K.C., A. (2012). *Predicting Sentence using N-Gram Language Model for Nepali Language* [Unpublished master's thesis]. Tribhuvan University, Central Department of Computer Science & Information Technology.

Khadka, B.R. (2020). *Predicting word using N-gram language model for Nepali language* [Unpublished master thesis]. Pokhara University, Nepal College of Information Technology.

Microsoft SQL Server 2005 Express Edition. (n.d.). https://www.microsoft.com/en-us/download/details.aspx?id=21844

Microsoft-visual-studio-2010-professional-free-download. (n.d.). https://www.freesoftwarefiles.com/development/microsoft-visual-studio-2010-professional-free-download/

Rani, A., Singh, U., & Goyal, V. (2014). Disambiguating Hindi words using N-gram smoothing models. *An International Journal of Engineering Sciences*, *10*, 26–29.

Yazdani, A., Reza, S., Golkar, & Kalhari. S.R.N. (2019). Words prediction based on N-gram model for free-text entry in electronic health records. *Health Information Science and Systems*, *7*(1), 6. https://doi.org/10.1007/s13755-019-0065-5

To cite this article [APA style, 7th edition]:
Khadka, B. R. (2022). The use of K-gram language model in predicting Nepali words. *Prithvi Academic Journal*, *5*, 46-58.