Int. j. inf. tecnol. (June 2022) 14(4):2105–2118
https://doi.org/10.1007/s41870-020-00503-y

ORIGINAL RESEARCH

Experimenting with factored language model and generalized back-off for *Hindi*

Arun R. Babhulgaonkar[1] · **Shefali P. Sonavane**[2]

Received: 1 February 2020 / Accepted: 28 July 2020 / Published online: 28 August 2020
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2020

Abstract Language modeling is a statistical technique to represent the text data in machine readable format. It finds the probability distribution of sequence of words present in the text. Language model estimates the likelihood of upcoming words in some spoken or written conversation. Markov assumption enables language model to predict the next word depending on previous n − 1 words, called as n-gram, in the sentence. Limitation of n-gram technique is that it utilizes only preceding words to predict the upcoming word. Factored language modeling is an extension to n-gram technique that facilitates to integrate grammatical and linguistic knowledge of the words such as number, gender, part-of-speech tag of the word, etc. in the model for predicting the next word. Back-off is a method to resort to less number of preceding words in case of unavailability of more words in contextual history. This research work finds the effect of various combinations of linguistic features and generalized back-off strategies on the upcoming word prediction capability of language model over *Hindi* language. The paper empirically compares the results obtained after utilizing linguistic features of *Hindi* words in factored language model against baseline n-gram technique. The language models are compared using perplexity metric. In summary, the factored language model with product combine strategy produces the lowest perplexity of 1.881235. It is about 50% less than traditional baseline trigram model.

Keywords Factored language model (FLM) · Generalized back-off · n-gram · Perplexity

✉ Arun R. Babhulgaonkar
arbabhulgaonkar@dbatu.ac.in

Shefali P. Sonavane
shefali.sonavane@walchandsangli.ac.in

[1] Dr. Babasaheb Ambedkar Technological University, Lonere, Maharashtra, India

[2] Walchand College of Engineering, Sangli, Maharashtra, India

1 Introduction

Hindi is the national and official language of India. According to https://www.Vistawide.com after English, Spanish and Mandarin, *Hindi* is the most natively spoken language. It is used by about 400 million people. It goes beyond 400 million if dialects of *Hindi* are considered which share the *Devenagari* script of *Hindi* language such as Marathi, Sanskrit, etc. Most of the government resolutions, documents, historical records, etc. are available in English which may not be understood by the villagers in India. This raises a need to develop an efficient automatic translation system from English to *Hindi*. Machine Translation is a technique used for translation of text in source natural language into target natural language. Language model is very essential component of a statistical machine translation (SMT) system. A language model finds the likelihood of words during some conversation in any natural language. It finds the tendency of words following other words in a natural language. The language model is also used to predict which word would be coming next depending on the previous words already appeared in the sentence [1]. The language model stores the probability of each possible upcoming word for the given context of words in the training data. For the given sequence of words in test data it predicts the word with highest probability as the next possible word. It is also used to assign a probability value to an entire sentence. If W $= w_1w_2w_3\ldots w_n$ is a sentence in a natural language then the language model technique estimates the probability p(W) of whole sentence using chain rule of probability and Markov assumption. Today, language models are used by Google in almost all online services for sentence completion while a user is trying to type some text. Language model restricts the search space and provides guidance to select next word hypothesis to complete the sentence. Apart from on words, language model can also be constructed on sequence of characters, signs or symbols.

This research work shows that selection of next word in a valid sentence in *Hindi* language does not depend merely on preceding words. Going beyond n-gram technique which uses surface form of words only, through experimentation it is demonstrated that linguistic knowledge of previous words, called as factors, also plays very crucial role for predicting upcoming words. As *Hindi* is morphologically a very rich language it provides many factors. The significance of this research work lies in two objectives. The first objective of the research work is to find the effective factors of words in *Hindi* language by using factored language modeling (FLM) technique. Experiments done confirms that FLM significantly improves the prediction capability and offer significant reduction in perplexity when compared to standard n-gram language models. Generalized back-off in FLM explores many options and dynamically selects the efficient lower order model when enough contextual history is not available for predicting the upcoming word. This gives rise the second objective of research as to find effective back-off strategy in FLM. Thus, significance of this research work is that it finds the effect of various combinations of linguistic features and generalized back-off strategies on the upcoming word prediction capability of language model over *Hindi* language. While doing translation from any other source language to *Hindi* language, due to structural difference between that source language and *Hindi* language, the order of words gets changed. Hence, it becomes very essential to find out the sequential order in which the correctly translated words must be placed. Here, the language model plays its role and statistically determines the correct order of words and generates a fluent translated output sentence. The results of the study will definitely empower the developer of a machine translation system to use FLM over *Hindi* language instead of using the baseline n-gram model to get better translation results. The results of experimentation will provide the developer of a machine translation system with information on how the factors of previous words affect the selection of next word. This study will provide information regarding which group of factors for the *Hindi* language is effective and may improve the translation quality. The results will enable him to think about utilization of other factors also for getting better translation performance. The results of experimentation done for generalized back-off during the study will help the developer of a machine translation system to select the effective back-off strategy for *Hindi* language.

2 Language modeling techniques

(A) Standard n-gram language modelling

The leading method of language modelling is n-gram where, n indicates the number of words used. n also indicates the degree of the language model. n-gram language modelling is a technique to capture the probability of how likely words are following each other. Ideally, a word appearing at current time step in the sentence may depend on any word appeared at previous time steps in the sentence. But, Markov assumption simplifies the conditional dependency of next word and suggests utilizing only limited backward context of words. Thus, standard n-gram language model utilizes previous n − 1 words only, called as contextual history *h,* for predicting the nth word. Mathematically, probability of next word w_n coming after the word sequence $w_1w_2w_3\ldots w_{n-2}w_{n-1}$ in the sentence is given as:

$$p(w_n|h) = p(w_n|w_1w_2w_3\ldots w_{n-2}w_{n-1}) \quad (1)$$

For example, if the contextual history is अरुण ने खत, then the probability of the next word being लिखा can be represented by using conditional probability as:

$$p(\text{लिखा}|\text{ अरुण ने खत}) \quad (2)$$

One way to calculate this probability is by using relative frequency counts of the construct. Count of how many times अरुण ने खत is appearing in corpus is obtained first and then count of how many times it is followed by word लिखा is obtained. Using maximum likelihood estimation (MLE) technique it is written as:

$$p(\text{लिखा}|\text{अरुण ने खत}) = \frac{C(\text{अरुण ने खत लिखा})}{C(\text{अरुण ने खत})} \quad (3)$$

Thus, in a sufficiently large corpus, the probability distribution over the sequence of n words present in the corpus can be obtained using MLE technique. In generalised form, MLE of nth word is represented as:

$$p(w_n/w_1,\ldots,w_{n-2},w_{n-1}) = \frac{C(w_1,\ldots w_{n-2},w_{n-1}w_n)}{C(w_1,\ldots,w_{n-2},w_{n-1})} \quad (4)$$

In Eq. 4, the n-gram probability is estimated by taking ratio of total observed frequency count of a particular sequence of n words and the total observed frequency count of the given prefix of n – 1 words for the nth word.

(B) Factored language model (FLM)

In any natural language a valid sentence is formed by keeping the words in that language in certain order by following the grammatical rules of the language. It is fact of any natural language that upcoming word in a sentence does not depend on preceding words only, but also depends on the grammatical category of preceding words in the sentence. Part-of-Speech, number, gender etc. of preceding words plays a very vital role in selection of next word in the sentence. For instance, in *Hindi*, 'लता गाना गाती है' is more correct than 'लता गाना गाता है' from grammatical point of view of *Hindi* language as gender of noun word 'लता' is feminine. This is due to the fact that unlike English, the verb in *Hindi* language changes according to the gender of the subject in the sentence.

n-gram language model uses only preceding words but FLM also utilizes grammatical and linguistic knowledge of preceding words for predicting the next word. In FLM, each word now can be seen as a set of k factors. Hence, tth word w_t in the sentence will be written as: $w_t = \{f_{t1}, f_{t2}, \ldots, f_{tk}\}$ where, $\{f_{t1}, f_{t2}, \ldots, f_{tk}\}$ is set of factors associated with the word. Each word itself is considered as a factor and other factors associated with the word may be any grammatical information of the word such as stem of the word, part-of-speech tag, morphological class or any other linguistic information of the word. For highly inflected languages it will be more useful as more and more factors of a word will be obtained in it. The factored representation is applicable to any language as semantic features available in that language can be considered as factors. Thus, FLM calculates probability distribution of sequence of words and its associated factors as shown in Eq. 5.

$$p(f_{1:t}^{1:k}) \tag{5}$$

where, $f_{1:t}^{1:k}$ is any factor from 1 to k of any word from 1 to t. Probability of any factor f from 1 to k of any tth word can be obtained by depending on all the factors used in the contextual history of the factor. Using n-gram formalism, it is shown as:

$$p(f_t^{1:k} | f_{t-n+1:t-1}^{1:k}) \tag{6}$$

The probability of any particular kth factor f_t^k of tth word can be obtained using all the remaining factors $f_t^{1:k-1}$ of that word and all the factors of all the words $f_{t-n+1:t-1}^{1:k}$ in the history as shown in Eq. 7:

$$\prod_k p(f_t^k | f_t^{1:k-1} \ldots f_{t-n+1:t-1}^{1:k}) \tag{7}$$

Equation 7 shows a single possible ordering among many orderings of the factors by chain rule. Apart from this, any set of factors in any order can be selected for finding the probability of the next factor. k factors of a word can be permuted in k! ways and totally 2^{nk} subsets of factors can be obtained. Here, n is the number of words and k is the number of factors associated with each word totally n*k factors. This way FLM opens up $k!2^{nk}$ possibilities for language modelling options in addition to standard n-gram model [4]. Standard n-gram model can be treated as the simplest case of the FLM in which only one factor i.e. word itself, is considered. Same is the case with class-based model where class is considered as a factor and it depends only on the previous class. Figure 1 shows generalised representation of FLM.

The morphologically rich language like *Hindi* makes many linguistic and grammatical features available to be used as a factor in factored language modelling. For this research work following factors in *Hindi* are used:

1. Word (शब्द)
2. Part-of-speech (POS) (शब्द भेद)
3. Gender (लिंग)
4. Number (वचन)
5. Stem of the word (मूलशब्द)

Figure 2 shows the graphical representation of the FLM for a word in *Hindi* language. Equation 8 shows same FLM model in conditional probability form:

$$p(W_t | W_{t-1}, G_{t-1}, N_{t-1}, P_{t-1}, W_{t-2}) \tag{8}$$

where, W_t is word factor, G_t is the word's gender tag factor, N_t is the word's number tag factor, P_t is the word's part-of-speech tag factor. Here, probability of next word factor is conditioned on immediate previous word's number, gender and POS tag factors along with previous two words. Like this probability of next factor can be obtained by taking any useful combination of factors present in the contextual history of the word. For instance, one more combination is:

$$p(W_t | W_{t-1}, W_{t-2}, G_{t-1}, G_{t-2}, P_{t-1}, P_{t-2}) \tag{9}$$

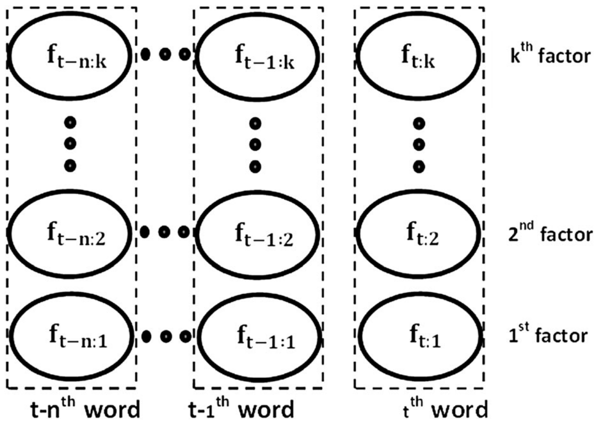

Fig. 1 Graphical representation of factored language model

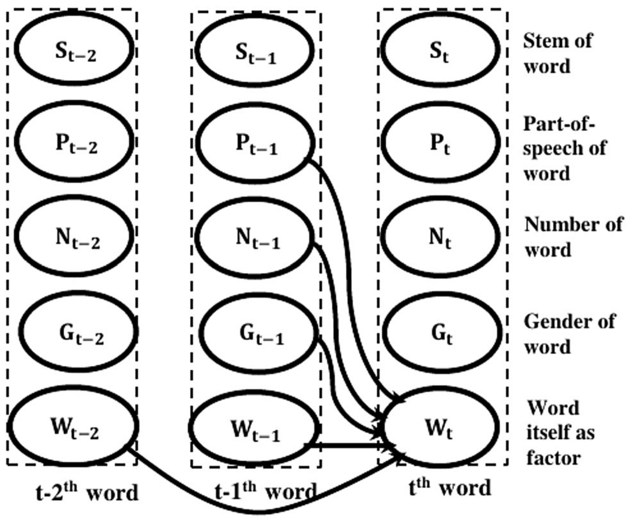

Fig. 2 FLM for a word

In this model, the word factor W_t is used as child and the factors $W_{t-1}, W_{t-2}, G_{t-1}, G_{t-2}, P_{t-1}$ and P_{t-2} are used as the parents. This FLM model is same as word only trigram model but along with words previous two word's POS tags and gender tags are also considered as parents to predict next word factor.

3 Back-off

Back-off is the method of obtaining probability distribution of n-grams when the higher order n-grams do not have sufficient probability mass. In other word, back-off suggests that, when data is not sufficient to estimate a high order n-gram conditional probability, estimate only a portion of the table and construct the remaining table using a lower order n-gram model. For example, when the trigram $p(W_t|W_{t-1}, W_{t-2})$ count is not sufficient, move down to the bigram $p(W_t|W_{t-1})$. This process is then recursively applied down up to unigram.

The back-off model for trigram $P_{BO}(W_t|W_{t-1}, W_{t-2})$ is defined as given in Eq. 10:

$$P_{BO}(W_t|W_{t-1}, W_{t-2}) = \begin{cases} d_{N(w_t,w_{t-1},w_{t-2})} P_{ML}(W_t|W_{t-1}, W_{t-2}) \\ \alpha(W_{t-1}, W_{t-2}) P_{BO}(W_t|W_{t-1}) \end{cases} \tag{10}$$

It says that, trigram distribution is used if its frequency count is greater than threshold value τ_3 otherwise go to the bigram model after applying the discount function $d_{N(w_t,w_{t-1},w_{t-2})}$ on the trigram distribution. $d_{N(w_t,w_{t-1},w_{t-2})}$ is a number ranging from 0 to 1. For discounting, different smoothing techniques can be used such as Good Turing smoothing, Kneser–Ney smoothing, Natural discounting, etc. [2]. The quantity $\alpha(w_{t-1}, w_{t-2})$ ensures that the sum of entire distribution remains equal to unity after applying back-off.

Figure 3 shows the back-off right from 4-gram to the unigram distribution. In this back-off model, if 4-gram sequences are not in sufficient amount then back-off to 3-gram sequences. If 3-gram sequences are also not enough then go for bigram count and same way up to unigram distribution.

3.1 Generalized back-off for FLM

In standard n-gram language model, back-off procedure drops the word that is most distant in the contextual history of the next word. Successively next most distant word is then dropped and this way the procedure goes on till the unique neighbouring preceding word of the next word is dropped and a unigram distribution is obtained. In the graphical representation shown in Fig. 3, first the most distant parent node W_{t-3} is dropped. Then the next distant node W_{t-2} is dropped and so on till the immediate parent node W_{t-1} is dropped. The graphical representation of the back-off procedure shown in Fig. 3 is called as back-off graph. It starts from a node where all three previous words are present and steps down to the node where only single immediate preceding word remains and then last unigram node. In FLM, due to availability of a lot of factors and possibility of various combinations of the factors in the contextual history of the word, many options are available for dropping during the back-off procedure.

For example, if for a word in *Hindi* language the FLM is;

$$p(W|W_1, N_1, P_1) \tag{11}$$

Then for this 4-gram FLM all the possible back-off paths will be as shown in Fig. 4.

Each path in the back-off graph is a distinct back-off model. Hence, which back-off path should be selected is a decision problem.

Figure 5 shows only specific back-off paths in the back-off graph given in Fig. 4. In this graph, intermediate node

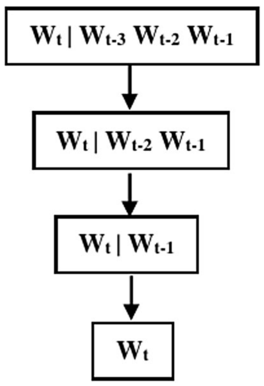

Fig. 3 Back-off process in a 4-gram model

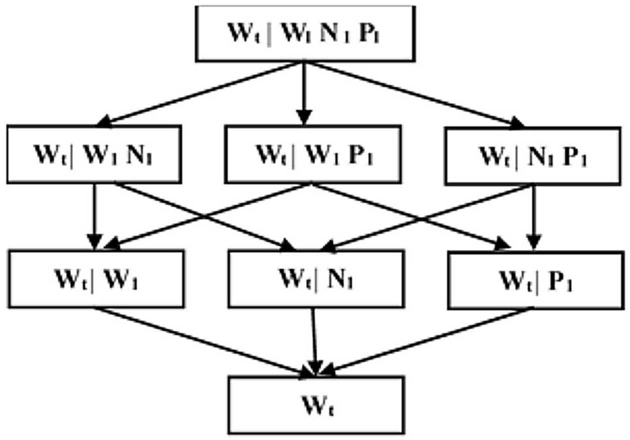

Fig. 4 Back-off graph in FLM

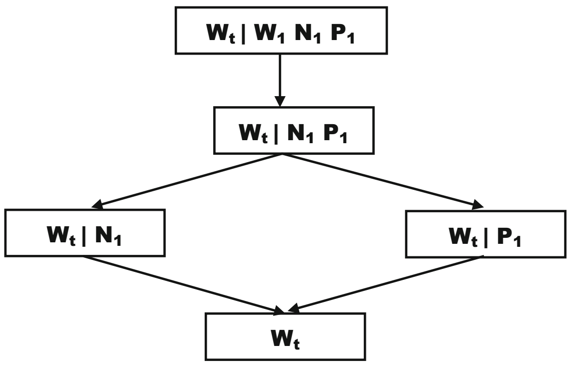

Fig. 5 Back-off path in FLM

$W_t|N_1P_1$ is obtained by dropping W_1 factor in the root node $W_t|W_1N_1P_1$. Now, two options are available for drooping i.e. N_1 and P_1. The selection of a factor to be dropped is a decision problem. The generalised back-off method solves this decision dilemma.

Generalized Back-off for FLM is generalization of standard back-off in language modelling [3]. In generalized back-off for FLM, while backing off from higher to lower order language model, instead of selecting a fixed path, multiple paths are selected dynamically at run time according to a preselected threshold of probability for a particular factor. There are two approaches for selecting the multiple back-off paths. In first approach, only one path is selected at a time for a set of factors but it can be changed as any other set of factors is required. It means as the set of factors is changed the back-off path can also be changed. In second approach, for a given set of factors multiple back-off paths are used simultaneously at runtime to obtain the probability.

Probabilities in generalized back-off (p_{GBO}) for FLM are obtained using Eq. 12 [3].

$$p_{GBO}(f|f_1,f_2,f_3) = \begin{cases} d_{N(f,f_1,f_2,f_3)}p_{ML}(f|f_1,f_2,f_3) & \text{if} N(f,f_1,f_2,f_3) > \tau_4 \\ \alpha(f_1,f_2,f_3)g(f,f_1,f_2,f_3) & \text{otherwise} \end{cases} \tag{12}$$

where, p_{ML} is maximum likelihood of factor counts which is given as:

$$p_{ML}(f|\ f_1,f_2,f_3) = \frac{N(f,f_1,f_2,f_3)}{N(f,f_1,f_2)}$$

It says that, find the maximum likelihood estimate of given set of factors and compare it with the predefined threshold value τ specified by the user. If it is greater than threshold, then a hit is occurred and the probability is retained otherwise back-off is applied. $g(f,f_1,f_2,f_3)$ is non-negative back-off distribution function. $\alpha(\ f_1,f_2,f_3)$ ensures that sum of entire distribution remains equal to unity. There are many possible ways how the function $g(f,f_1,f_2,f_3)$ can be calculated such as max normalized counts, min normalized counts, geometric mean, weighted mean, average mean and their variations. Each way of calculation of $g(f,f_1,f_2,f_3)$ results in a different back-off strategy. In the remaining part of this section, all such back-off strategies are presented as given in tutorial [4].

(1) Fixed back-off path

In fixed back-off path strategy, the path is fixed for all the process. It is initially defined by the user at start in the specification and remains same for all set of factors at all the time. User can define any one of the following fixed back-off path for a 4-gram model.

$$\begin{aligned} & g(f,f_1,f_2,f_3) = P_{BO}(f\ |f_1,f_2) \\ \text{or } & g(f,f_1,f_2,f_3) = P_{BO}(f\ |f_1,f_3) \\ \text{or } & g(f,f_1,f_2,f_3) = P_{BO}(f\ |f_2,f_3) \end{aligned} \tag{13}$$

(2) Max counts

As name indicates, the back-off node is selected by comparing the counts of all the possible back-off models and the model for which count is maximum is selected. This gives rise the opportunity to select different back-off path for each instance. This strategy prefers statistical estimation properties than the statistical predictability of the back-off models while selecting back-off path. For a 4-gram model, it may be defined as in Eq. 14:

$$g(f,f_1,f_2,f_3) = P_{GBO}(f|f_{a_1},f_{a_2}) \tag{14}$$

where,

$$(a_1,a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} N(f,f_{b_1},f_{b_2})$$

In this, count of the pairs (f_1,f_2), $(f_1,f_3)$and (f_2,f_3) is obtained and the pair for which the count is maximum is selected for back-off path purpose.

(3) Max normalized counts

This strategy is similar to max counts strategy. Only difference is, in this the normalized counts are used instead of normal counts for selecting the back-off node. The back-off node is selected by comparing the maximum likelihood estimates of all the possible back-off models and the model for which estimate is maximum is selected. This strategy prefers statistical predictability than the statistical estimation properties of the back-off models while selecting back-off path. For a 4-gram model, it may be defined as in Eq. 15:

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{15}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} \frac{N(f, f_{b_1}, f_{b_2})}{N(f_{b_1}, f_{b_2})} = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} P_{ML}(f\ |f_{b_1}, f_{b_2})$$

(4) Max num-words normalized counts

This strategy is similar to max counts strategy. Only difference is that, for the present set of parent values, the number of possible factor values in the training set are used for normalization purpose. The possible factor values for present parent context f_1, f_2, f_3 in a 4-gram model are expressed as $\{f: N(f, f_1, f_2, f_3) > 0\}$. The cardinality of this set represents the number of possible upcoming factors for the present context:

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{16}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} \frac{N(f, f_{b_1}, f_{b_2})}{\{f\ : N(f, f_{b_1}, f_{b_2}) > 0\}}$$

(5) Max back-off-graph node probability

In this strategy, the back-off node is selected by comparing the probability of all the possible back-off models and the model for which probability is maximum is selected. This is similar to fixed back-off strategy but instead of selecting back-off node initially, it is selected dynamically. For a 4-gram model, it may be defined as in Eq. 17:

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{17}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} P_{GBO}(f|f_{b_1}, f_{b_2})$$

(6) Max product-of-cardinality normalized counts

Underlying random variables cardinality product is used for normalization in this strategy. The possible values of random variable constitute its cardinality. The cardinality F is represented as:

$$|F| \triangleq \{f : N(f) > 0\} \tag{18}$$

For a 4-gram model, it may be defined as in Eq. 19:

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{19}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} \frac{N(f, f_{b_1}, f_{b_2})}{|F||F_{b_1}||F_{b_2}|}$$

(7) Max sum-of-cardinality normalized counts

In this strategy, instead of using the product of underlying random variables cardinality, their sum is used for normalization. The cardinality F is represented as:

$$|F| \triangleq \{f : N(f) > 0\}$$

For a 4-gram model, it may be defined as in Eq. 20.

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{20}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} \frac{N(f, f_{b_1}, f_{b_2})}{|F| + |F_{b_1}| + |F_{b_2}|}$$

(8) Max sum-of-log-cardinality normalized counts

This is similar to max sum-of-cardinality normalized counts strategy. The difference is that, for normalization purpose in this strategy instead of using the sum of underlying random variables cardinality, the sum of natural logarithm of the cardinalities is used. The cardinality F is represented as:

$$|F| \triangleq \{f : N(f) > 0\}$$

For a 4-gram model, it may be defined as in Eq. 21.

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{21}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmax}} \frac{N(f, f_{b_1}, f_{b_2})}{\ln|F| + \ln|F_{b_1}| + \ln|F_{b_2}|}$$

(9) Min counts

In contrast to the max strategies discussed earlier, the back-off node is selected by comparing the counts of all the possible back-off models and the model for which count is minimum is selected. This gives rise the opportunity to select different back-off path for each instance. This strategy prefers statistical estimation properties than the statistical predictability of the back-off models while selecting back-off path. For a 4-gram model, it may be defined as in Eq. 22:

$$g(f, f_1, f_2, f_3) = P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{22}$$

where,

$$(a_1, a_2) = \underset{(b_1,b_2)\in\{(1,2),(1,3),(2,3)\}}{\operatorname{argmin}} N(f, f_{b_1}, f_{b_2})$$

In this, count of the pairs (f_1, f_2), $(f_1, f_3)$and (f_2, f_3) is obtained and the pair for which the count is minimum is selected for back-off path purpose.

Similar to max strategies, there are six more min strategies as listed in Table 2 in section six given below. In these strategies, the counts of all the possible back-off models are obtained using various mathematical operations as given for max strategies and the back-off node for which the count is minimum is selected.

In the following strategies, instead of using a single back-off path, multiple paths are used to obtain the final probability score. These strategies are different from above discussed min or max strategies. In these strategies, instead of finding minimum or maximum counts of the back-off probable candidates, the mathematical operation of all the back-off candidate probabilities is calculated for finding back-off model.

(10) Sum

The sum of all the back-off candidate probabilities is calculated for finding back-off model. For a 4-gram model it may be defined as in Eq. 23:

$$g(f, f_1, f_2, f_3) = \sum_{(a_1,a_2)\in\{(1,2),(1,3),(2,3)\}} P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{23}$$

(11) Average (arithmetic mean)

The arithmetic mean of all the back-off candidate probabilities is calculated for finding back-off model. For a 4-gram model, it may be defined as in Eq. 24:

$$g(f, f_1, f_2, f_3) = \frac{1}{3} \sum_{(a_1,a_2)\in\{(1,2),(1,3),(2,3)\}} P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{24}$$

(12) Product

The product of all the back-off candidate probabilities is calculated for finding back-off model. For a 4-gram model, it may be defined as in Eq. 25:

$$g(f, f_1, f_2, f_3) = \prod_{(a_1,a_2)\in\{(1,2),(1,3),(2,3)\}} P_{GBO}(f|f_{a_1}, f_{a_2}) \tag{25}$$

(13) Geometric mean

The geometric mean of all the back-off candidate probabilities is calculated for finding back-off model. For a 4-gram model, it may be defined as in Eq. 26:

$$g(f, f_1, f_2, f_3) = \left(\prod_{(a_1,a_2)\in\{(1,2),(1,3),(2,3)\}} P_{GBO}(f|f_{a_1}, f_{a_2}) \right)^{1/3} \tag{26}$$

(14) Weighted mean

In this strategy, different weights using numerical values are assigned by the user to each available back-off node and the weighted mean of all the back-off candidate node probabilities is calculated for finding back-off model. For a 4-gram model, it may be defined as in Eq. 27.

$$g(f, f_1, f_2, f_3) = \prod_{(a_1,a_2)\in\{(1,2),(1,3),(2,3)\}} P_{GBO}^{\gamma_{a_1,a_2}}(f|f_{a_1}, f_{a_2}) \tag{27}$$

where, γ_{a_1,a_2} represents the weights assigned to the factors a_1 and a_2 by the user.

4 Related work

The initial use of FLMs reported were improvement of perplexity of language models on the WSJ corpus and the Arabic Callhome corpus. After that the models were used in various applications of NLP such as language modelling component of phrase based machine translation (PBMT), speech recognition. In paper [3], authors demonstrated how use of generalized parallel back-off (GPB) along with FLM improves the results when compared to n-gram models. Kirchhoff et al. in their tutorial described in detail the implementation of FLM using SRILM toolkit [4]. While in standard baseline language model a back-off path is selected manually and remain fixed, in their tutorial on FLM the authors proposed many ways for finding the back-off paths, which they called generalized back-off paths, based on probability and statistics of occurrence of data. They used geometric mean, min/max counts and weighted mean for back-off purpose. FLMs are always used in Slavic and Arabic languages. It is also used widely in Russian languages [7, 8]. It is observed that FLMs are also effectively used in other morphologically rich languages like Amharic [9] and Turkish [10]. In paper [10], authors showed that even with only limited size of available corpora for training, the FLM based language models for Turkish language are far better than traditional n-gram models. These results proved relevant in speech recognition task for some specific domains where data scarcity was a problem. In paper [11], authors used FLM for Portuguese text generation. In paper [12], authors used linguistic factors of the word in FLM for generating a Romanian language model. The FLMs are also used for N-best list rescoring in paper [13]. Authors described the use of a POS tag of a word in FLM for rescoring the sentences. They used first pass based on morphemes and second pass based on FLM models which further improved the results. In paper [14], authors claimed that morpheme based models need higher ordered models as compared to word based models to get comparable performance. In paper [15], the authors discussed morpheme based FLM which is a methodology for language modelling to combine both FLM approach and morphological decomposition of words. They obtained approximately 2.5% reduction in Word Error Rate (WER) for speech recognition task against a traditional full words system. In paper [16], the authors proposed that the morpheme based language model is observed to be more effective for large vocabulary continuous speech recognition (LVCSR) than the whole word language models. In paper [17], the authors used POS tags as syntactic features. They proposed that for speech recognition task, FLMs outperformed traditional trigram language models regarding mixed error rate and perplexity. They extracted syntactic and semantic features from code-switching text data and integrated it into FLM. In paper [18], the authors described their investigations of factors in code-switching language modelling. They considered Brown word clusters, POS tags, individual words, open class word clusters as factors in FLM and evaluated perplexity of the model. The grouping of these factors resulted in best perplexity. FLM developed using these factors reduced the perplexity by 10.8% on SEAME evaluation set and the error rate by 3.4%. In paper [19] also, authors applied FLM for speech recognition task.

Recently, Ganji and Sinha used FLM for recognizing the code-switched data containing mixing of English and *Hindi* (Hinglish) text [20]. For constructing FLM they added a novel factor named as CS-factor along with POS tag factor. CS-factor designed by them differentiate the word at which the code is shifted from *Hindi* language to English language. They obtained a significant improvement in perplexity for Hinglish data. Gregor and Zdravko proposed a novel idea of back-off that is context-dependent and claimed that it performs better than simple back-off method for speech recognition [21]. They constructed FLM by using factors obtained from morph-syntactically tagged data.

The real application of FLM is in statistical machine translation. In [22], author first utilized FLM for machine translation application. Nair et al. proposed a hybrid MT system based on singular, plural, case, gender knowledge obtained from declensions of noun and adjectives [23]. They also used morphological analyzer and POS tagger to annotate the words of a sentence. Ramanathan et al. used FMT for English to *Hindi* translation by including the corresponding suffixes and semantic relations of words in English sentences by aligning them with case markers and inflections of words in *Hindi* language [24]. During translation, many forms of a *Hindi* word are possible for an English word. But, in the given training data some of the forms may be missing. Hence, Sreelekha and Bhattacaryya improved the performance of English to *Hindi* translations by injecting missing morphological forms of words as factors in the target *Hindi* language corpus [25]. Through this they also reduced the out of vocabulary problem and improved the fluency of output. Patel et al. used FMT for many English to Indian languages including *Hindi* [26]. They first restructured the source English language sentences according to target language to tackle the structural divergence problem of languages. They also separated the suffixes from words and used stem of words as factors in FMT. This way they handled the morphological divergence problem. Jaya and Gupta also claimed the improvement in the quality of English to *Hindi* translation by augmenting the parallel corpus with morphological variations for *Hindi* noun and adjective [27]. Kumar et al. utilized part-of-speech tags, lemma, dependency information and syntactic information as factors for English-Tamil translation and claimed improvement in performance [28]. Dungarwal et al. [29] used case, number and Tree Adjoining Grammar information factors for improving English–*Hindi* translation quality in factored machine translation. Sachdeva et al. used POS tag factor in FMT for *Hindi* to English machine translation [30].

Researchers are now trying to integrate factored approach in Neural Machine Translation (NMT). García-Martínez et al. utilized morphological and many other factors of words in NMT [31]. They tried to manage the larger vocabulary problem of NMT. Wilken and Matusov extended the work of García-Martínez et al. by exploring the usefulness of factors beyond prediction of word lemmas and its inflections [32]. Hokamp used word level features for ensemble of NMT systems and achieved comparable results in both quality estimation (QE) & automatic post editing (APE) tasks [33].

It can be observed that, many different kinds linguistic factors of *Hindi* language words are used by many researchers. Various factors contribute differently in the task of prediction of upcoming word in a *Hindi* sentence. This motivates to find out the best factors of a *Hindi* word that influence a lot on upcoming word in a *Hindi* sentence. Many back-off models are proposed in the literature [3, 21, 34, 35]. This research work compares basic back-off model and generalized back-off model in context of *Hindi* language.

5 Methodology

5.1 Selection of factors

For using FLM to natural language processing task, first some effective and informative factors in *Hindi* language are identified. Then the effective FLM that produces lowest perplexity is selected out of many possible models. For training purpose, the corpus is pre-processed by adding number, gender and POS tag factors in the text. The significance and reason behind selection of these factors especially for *Hindi* language can be justified as:

(1) Gender factor

In *Hindi* language, next word depends on gender of words in contextual history of next word. For instance, in the *Hindi* sentence "सीता पढ़ने जाती है", subject of the sentence is noun word 'सीता'. Here, gender of the subject is female hence verb used is 'जाती'. But, if the subject is changed to a noun having male gender as 'राम' then verb 'जाता' must be used instead of 'जाती'. Thus, gender of previous words determines the upcoming word in the sentence.

(2) Number factor

In *Hindi* language, next word depends on the grammatical category 'number' related to words in contextual history of the next word. For instance, in the *Hindi* sentence "राम ने बहुत आम खाये", adjective used is 'बहुत' that is plural in nature hence the verb used is 'खाये'. But, if the adjective is changed by the word 'एक' then verb 'खाया' must be used to construct a correct *Hindi* sentence. Thus, grammatical category 'number' of previous words determines the upcoming word in *Hindi* sentence.

(3) Part-of-speech tag factor

In *Hindi* language, next word depends on part-of-speech tag of words in the contextual history of the next word. Occurrence of words in sequence of a valid *Hindi* language sentence is governed by certain rules regarding part-of-speech tags of words in the sentence. Such rules make prediction of next word easy as they give a high probability value to words with certain part-of-speech tag. Some such rules in *Hindi* language are given below.

Rule 1	Adjective is followed by noun with high probability For instance: सुरेश एक सच्चा दोस्त है. Here, adjective 'सच्चा' is followed by noun 'दोस्त'.
Rule 2	The reflexive pronoun is followed by noun with high probability. For instance: सुरेश अपने गांव गया है. Here, reflexive pronoun 'अपने' is followed by noun 'गांव'.
Rule 3	Noun is followed by verb or noun with high probability. For instance: (1) सुरेश सोने जा रहा है. (2) सुरेश युद्ध हार गया. Here, noun 'सोने' is followed by verb 'जा' and noun 'सुरेश' is followed by noun 'युद्ध'.
Rule 4	Main verb tag is followed by auxiliary verb with high probability. For instance: सुरेश सोने जा रहा है. Here, main verb 'जा' is followed by auxiliary verb 'रहा'.

Thus, while predicting next word, using part-of-speech tags the search space is restricted to a limited set of words with a certain part-of-speech tag only instead of searching all words in the vocabulary of the language.

5.2 Pre-processing of data

While working with FLM, pre-processing is first and most important step that includes identification of factors to be used and determination of possible values of each of the factor. In this research, factored language models are trained using *Hindi* language text having totally 21,835 factored words. Trained models are tested using a test corpus having 2144 words. During pre-processing, each and every word present in the corpus is annotated with all the factor values. Thus, every word is now considered as a set of factors. A grammatical factor and its associated value for that word are separated by '-' character. Consecutive factors associated with a word are differentiated by ':' character. Special tag symbols like<s>is used to mark beginning of each sentence and</s>is used to mark end of each sentence.

For example, if the *Hindi* sentence is:

"रेडियो और दूरदर्शन महत्वपूर्ण मौसम सूचनाओं और कृषकों से संबंधति सूचनाओं जैसी अन्य जानकारयिों का प्रसार करते हैं।"

After inserting factors, it will become:

<s>W-रेडियो: G-पुल्लगि: N-एकवचन: P-नाम: S-<NULL> W-और: G-<NULL>: N-<NULL>: P-समुच्चयबोधकअव्यय: S-<NULL> W-दूरदर्शन: G-पुल्लगि: N-एकवचन: P-नाम: S-<NULL> W-महत्वपूर्ण:G-पुल्लगि:N-एकवचन:P-वशिषण:S-<NULL> W-मौसम: G-पुल्लगि: N-एकवचन: P-नाम:S-<NULL>□□□</s>

where, W is word itself, G is gender, N is number, P is part-of-speech and S is stem factor of the word.

5.3 Experimentation

Factored language models are constructed using freely available SRILM language modelling toolkit [5, 6]. SRILM language modeling toolkit supports both generalized back-off and FLMs. Essentially, the SRILM provides a way to implement graphical models like syntax in terms of code. The code syntax can contain the factor to be used as child node and other required factors as its parent node. It also provides a way to specify each possible node that can be dropped to determine the back-off path in the back-off graph. Different options can be specified at each node for smoothing and threshold value required for back-off decision making. The SRILM accepts a factored language data file and produces one count file containing counts of each n-gram and one language model file showing probabilities of n-grams. The language model is generated as per the guidelines given in the language model specification file. SRILM provides various methods of smoothing such as Kneser–Ney smoothing, Good-Turing smoothing, Witten-Bell smoothing and Natural discounting [2].

5.3.1 *Experimentation number 1*

As mentioned earlier, apart from word itself four different factors such as POS tag, gender, number and word stem are considered for building FLMs. For identifying the factor that is effective in predicting upcoming word many FLM models are constructed. Various combinations of factors and discounting techniques are used for model generation. From the results, it is observed that among all the factors considered during experimentation, the 'number factor' associated with the previous word is more informative as compared to other factors for next word prediction. The sample example FLM specification program written during experimentation is:

5.3.2 *FLM specification program 1*

```
1. ## 1W1P Natural discounting FLM
2. W : 2 W(-1) P(-1) ~/FLM/models/1w1p.count.gz ~/FLM/models/1w1p.lm.gz 3
3. W1,P1      W1  ndiscount gtmin 0.1 interpolate
4.     P1      P1  ndiscount gtmin 0.1 interpolate
5.      0       0   ndiscount gtmin 0.1
```

where, ## indicates comment, W is a child node having two parent factors such as one previous word W (−1) and it's part-of-speech P (−1) factor. Here, minus sign indicates one word before the current word. Integer 3 at the end of line 2 indicates that three back-off nodes are used. But, here simple fixed back-off is used by dropping word factor first in line three and part-of-speech factor in line 4.

5.3.3 *Experimentation number 2*

Experiment 2 is for implementing generalized back-off in FLM. Once the best informative factor i.e. number factor, is identified in experimentation 1, it is combined with second best informative factor i.e. part-of-speech, and FLM models with generalized back-off are constructed. The sample example FLM models with generalized back-off specification programs written during experimentation are explained below.

5.3.4 *FLM specification program 2*

FLM specification program 2:

```
1. ## general back-off, combine sum FLM
2. W:3 W(-1)N(-1)P(-1) ~/knsum.count1.gz ~/knsum.lm1.gz 5
3. W1,N1,P1   W1 ukndiscount gtmin 0.1 interpolate
4.     N1,P1   N1,P1 ukndiscount gtmin 0.1 combine sum
               interpolate
5.     N1 N1   ukndiscount gtmin 0.1 kn-count-parent
               W1,N1,P1
6.     P1 P1   ukndiscount gtmin 0.1 kn-count-parent
               W1,N1,P1
7.     01 01   ukndiscount gtmin 0.1 kn-count-parent
               W1,N1,P1
```

Integer 5 at the end of line 2 indicates that five back-off nodes are used in the back-off graph. Line number 4 indicates the generalized back-off where simultaneously two nodes N1 and P1 are dropped from parent node W1, N1, P1 leading to two separate paths. This specification program produces "knsum.count1.gz" as the frequency count file and "knsum.lm1.gz" as the language model. The combine sum strategy instructs to take summation of the probabilities of these back-off paths to produce final probability of the model. Kneser–Ney discounting is used here with interpolation option. The gtmin parameter equal to 0.1 is threshold value used to restrict the n-grams with low probability count.

5.3.5 *FLM specification program 3*

FLM specification program 3:

```
1. ## general back-off, witten bell discounting combine
2. ## max strategy counts_no_norm FLM
3. W:3 W(-1)N(-1)P(-1) ~/wbmax.count1.gz ~/wbmax.lm1.gz 5
4. W1,N1,P1   W1 wbdiscount gtmin 0.1 interpolate
5.     N1,P1   N1,P1 wbdiscount gtmin 0.1 combine max
               Strategy counts_no_norm interpolate
6.     N1 N1   wbdiscount gtmin 0.1 kn-count-parent
                W1,N1,P1
7.     P1 P1   wbdiscount gtmin 0.1 kn-count-parent
               W1,N1,P1
8.     01 01   wbdiscount gtmin 0.1 kn-count-parent
               W1,N1,P1
```

Line number 5 indicates the generalized back-off where simultaneously two nodes N1 and P1 are dropped from parent node W1, N1, P1 leading to two separate paths. The combine max counts_no_norm strategy instructs to take the maximum of the probabilities of these back-off paths to produce the final probability of the model. Witten-Bell discounting is used here with interpolation option. Threshold parameter gtmin equal to 0.1 is kept same in all models generated to compare the models for different combine strategies with various discounting techniques.

To check the effect of generalized back-off, the standard n-gram models are also constructed. Many experiments are carried out by writing various scripts satisfying generalised back-off FLM specification format for testing different back-off paths and smoothing techniques.

6 Results

The language model can be evaluated in two different ways. One is intrinsic evaluation and other is extrinsic evaluation. For extrinsic evaluation, the obtained model is integrated in some NLP application and its performance is evaluated by measuring the effect of variations in the model on application output. In intrinsic evaluation, the language model is not applied to any NLP application but it is evaluated at its own performance. The intrinsic evaluation is a probabilistic approach and the performance is measured in it using a metric named as perplexity.

Table 1 Perplexity of language models without generalized back-off

Sr. no.	Model	Kneser–Ney discounting	Witten–Bell discounting	Natural discounting
1	Unigram p (W)	3.850182	3.917763	4.171215
2	Bigram p (W|W$_1$)	3.782754	3.930612	3.850802
3	Trigram p (W|W$_1$W$_2$)	3.763726	3.887517	3.808582
4	1 Previous gender factor p (W|G$_1$)	3.807609	3.873462	3.794812
5	1 Previous number factor p (W|N$_1$)	3.807105	3.873462	3.794812
6	1 Previous POS factor p (W|P$_1$)	3.800902	3.873462	3.794812
7	1 previous word and gender factor p (W|W$_1$G$_1$) W$_1$ dropped first without interpolate	2.671253	2.744799	2.689067
8	1 previous word and gender factor p (W|W$_1$G$_1$) W$_1$ dropped first with interpolate	2.886722	2.966200	2.689067
9	1 previous word and gender factor p (W|W$_1$G$_1$) G$_1$ dropped first without interpolate	4.131441	4.245190	4.158992
10	1 previous word and gender factor p (W|W$_1$G$_1$) G$_1$ dropped first with interpolate	4.386634	4.507408	4.160113
11	1 previous word and number factor p (W|W$_1$N$_1$) W$_1$ dropped first without interpolate	2.570084	2.640845	2.587223
12	1 previous word and number factor p (W|W$_1$N$_1$) W$_1$ dropped first with interpolate	2.781057	2.857626	2.587223
13	1 previous word and number factor p (W|W$_1$N$_1$) N$_1$ dropped first without interpolate	4.131441	4.245190	4.158992
14	1 previous word and number factor p (W|W$_1$N$_1$) N$_1$ dropped first with interpolate	4.386634	4.507408	4.160113
15	1 previous word and POS factor p (W|W$_1$P$_1$) W$_1$ dropped first without interpolate	2.885450	2.964893	2.904691
16	1 previous word and POS factor p (W|W$_1$P$_1$) W$_1$ dropped first with interpolate	3.111281	3.196941	2.904692
17	1 previous word and POS factor p (W|W$_1$P$_1$) P$_1$ dropped first without interpolate	4.131441	4.24519	4.158992
18	1 previous word and POS factor p (W|W$_1$P$_1$) P$_1$ dropped first with interpolate	4.386634	4.507408	4.160113
19	2 previous word factors p (W|W$_1$W$_2$) W$_1$ dropped first with interpolate	3.720315	3.842677	3.764653
20	2 previous word factors p (W|W$_1$W$_2$) W$_2$ dropped first without interpolate	4.119056	4.254533	4.168146
21	2 previous word factors p (W|W$_1$W$_2$) W$_2$ dropped first with interpolate	4.402426	4.547223	4.169269

Perplexity is measured by using entropy of the probability distribution. If the word sequence produces higher conditional probability, then the language model will have lower perplexity. Thus, Perplexity is inversely related to probability. For a given test corpus, if the language model is producing low perplexity then it is treated as a better model. For this research work the language models are evaluated by using perplexity metric.

The perplexity of standard n-gram models with degree 1, 2 and 3 is given in first three rows of the Table 1. It considers only previous words and does not consider any additional linguistic knowledge to predict the upcoming word. The results show that, as the degree of language model, i.e. number of previous word considered are increasing its perplexity is decreasing. It means to predict the upcoming word correctly more and more previous words are required.

The perplexity obtained on test corpus for the FLM without generalized back-off models is given in Table 1 from row 4 to row 21. The results in Table 1 are compared

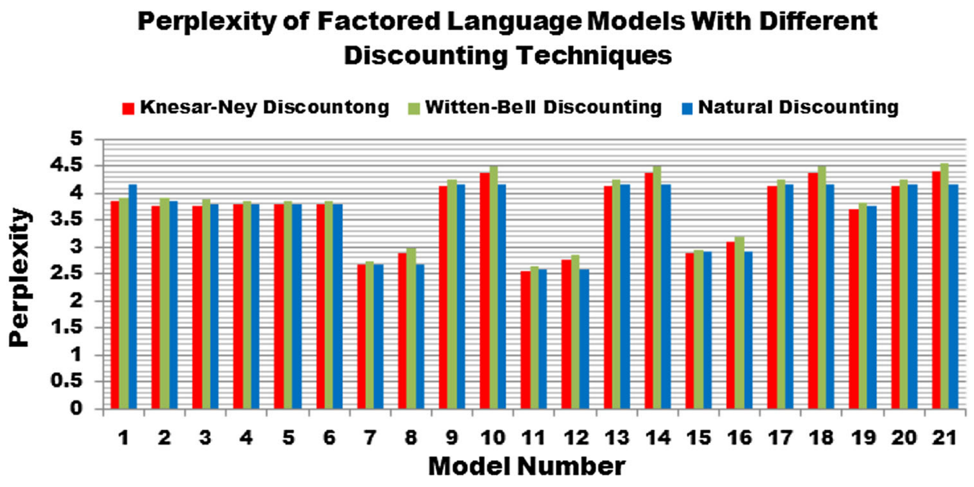

Fig. 6 Graph showing perplexity of language models without generalized back-off

Table 2 Perplexity of factored language models with generalized back-off using various combine strategies

Sr. no.	Combine strategy	Kneser–Ney discounting	Witten–Bell discounting	Natural discounting
1	Max normalized counts	4.771378	4.552701	5.309953
2	Max counts	4.755718	4.552701	5.309953
3	Max num-words normalized counts	4.776375	4.552701	5.309953
4	Max product-of-cardinality normalized counts	4.776375	4.552701	5.309953
5	Max sum-of-cardinality normalized counts	4.776375	4.552701	5.309953
6	Max sum-of-log-cardinality normalized counts	4.776375	4.552701	5.309953
7	Max back-off-graph node probability	4.776375	4.552701	5.309953
8	Sum	5.555569	5.742834	5.309953
9	Arithmetic mean	4.765860	4.552703	5.309957
10	Product	3.474772	1.982190	1.881235
11	Geometric mean	4.765773	4.552701	5.309953
12	Weighted mean	4.428558	4.324853	5.309957
13	Min normalized counts	4.760446	4.552701	5.309953
14	Min counts	4.755718	4.552701	4.755718
15	Min num-words normalized counts	4.755718	4.552701	4.755718
16	Min product-of-cardinality normalized counts	4.755718	4.552701	4.755718
17	Min sum-of-cardinality normalized counts	4.755718	4.552701	4.755718
18	Min sum-of-log-cardinality normalized counts	4.755718	4.552701	4.755718
19	Min back-off-graph node probability	4.755718	4.552701	4.755718

graphically in Fig. 6 where x axis shows the serial number of the language model in Table 1 and y axis is showing its perplexity. In this experimentation, the factors are dropped without using generalized back-off. FLM models produce better results than standard baseline n-gram language models. The perplexity is decreased from 3.763726 for baseline trigram model to 2.570084 for model p $(W|W_1N_1)$ when W_1 is dropped first without interpolate with Kneser–Ney discounting. Here, W_1 is immediate previous one word and N_1 is immediate previous word's number factor.

The perplexity obtained for various FLMs with generalized back-off using different combine strategies is listed in Table 2. The results in Table 2 are compared graphically in Fig. 7 where x axis shows the serial number of the strategy in Table 2 and y axis shows its perplexity. It is observed that, for available test corpus, the perplexity of FLM with product combine strategy is lowest as compared to other combine strategies. Out of the three discounting techniques used, Natural discounting technique with product combine strategy produces the lowest perplexity in all the models generated that is 1.881235. It is about 50% less than trigram baseline n-gram language model. In all developed FLMs, interpolation is used to estimate probabilities at the back-off node. For all discounting approaches

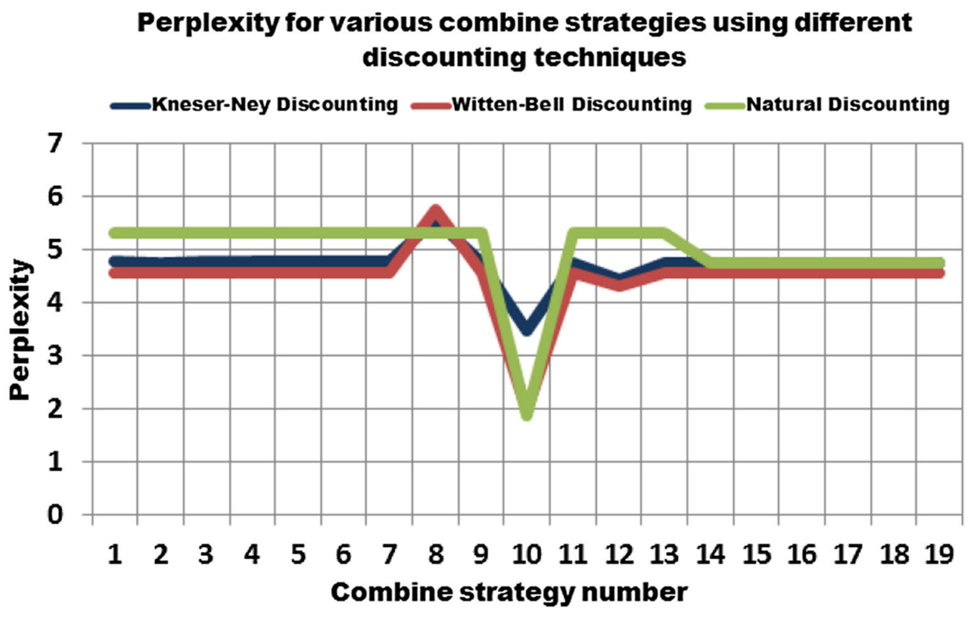

Fig. 7 Perplexity of FLM with generalized back-off using various combine strategies

same gtmin of 0.1 is used for training of all FLMs. As FLM utilizes grammatical and linguistic knowledge of the preceding words for predicting the next word, this more information adds more robustness in the model.

7 Conclusion and future scope

The generalized back-off is the generalization of back-off in baseline n-gram language modelling. The generalized back-off gives opportunity to consider all possible back-off paths simultaneously using various strategies and selects the most appropriate. As compared to n-gram model, FLM predicts the next word more correctly because of availability of additional knowledge of grammatical and lexical features of the words in the language. It is observed that Natural discounting technique with product combine strategy produces the lowest perplexity of 1.881235 and it is about 50% less than trigram word only language model. The difficulty to work with the FLM is huge pre-processing of text as every word in the corpus must be annotated by all the factors. Also, for FLM with many factors, the availability of many back-off paths increases the computational complexity of the model. In future work, some more linguistic features of *Hindi* language can be included in FLM and effective generalized back-off strategy can be identified. Combinations of available factors and levels of back-off graph may also be increased to check whether it can further improve the perplexity or not.

References

1. Rosenfeld R (2000) Two decades of statistical language modeling: where do we go from here? In: Proceedings of the 2000 IEEE international conference, vol 88, no 8, pp 1270–1278. https://doi.org/10.1109/5.880083
2. Chen SF, Goodman J (1996) An empirical study of smoothing techniques for language modelling. In: Proceedings of the thirty-fourth annual meeting of the association for computational linguistics, San Francisco, pp 310–318. https://doi.org/10.3115/981863.981904
3. Bilmes JA, Kirchhoff K (2003) Factored language models and generalized parallel back-off. In: Proceedings of the HLT/NAAC, pp 4–6. https://www.aclweb.org/anthology/N03-2002.pdf, https://doi.org/10.3115/1073483.1073485
4. Kirchhoff K, Bilmes J, Duh K (2008) Factored language models tutorial. University of Washington, Washington
5. Stolcke A (2002) SRILM—an extensible language modeling toolkit. In: Proceedings of the 2002 international conference on spoken language processing, Denver, Colorado
6. Stolcke A, Wheng J, Wang W, Abrash V (2011) SRILM at sixteen: update and outlook. In: Proceedings of the 2011 IEEE automatic speech recognition and understanding workshop, Waikoloa
7. Vazhenina D, Markov K (2013) Factored language modeling for Russian LVCSR. In: Proceedings of the international joint conference on awareness science and technology and ubi-Media computing (iCAST-UMEDIA), Aizuwakamatsu, pp 205–211. https://doi.org/10.1109/icawst.2013.6765434
8. Kipyatkova I, Karpov A (2014) Study of morphological factors of factored language models for Russian ASR. In: Proceedings of the international conference on speech and computer (SPECOM), Novi Sad, pp 451–458. https://doi.org/10.1007/978-3-319-11581-8_56
9. Tachbelie MY, Abate ST, Menzel W (2009) Morpheme-based and factored language modeling for Amharic speech recognition.

In: Proceedings of the 4th conference on human language technology: challenges for computer science and linguistics, Poznan, pp 82–93. https://doi.org/10.1007/978-3-642-20095-3_8
10. Sak H, Saraçlar M, Güngör T (2010) Morphology-based and subword language modeling for Turkish speech recognition. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Dallas, pp 5402–5405. https://doi.org/10.1109/icassp.2010.5494927
11. Novais E, Ivandré P (2012) Portuguese text generation using factored language models. J Braz Comput Soc 19(2):135–146. https://doi.org/10.1007/s13173-012-0095-1
12. Lazăr M, Militaru D (2013) A Romanian language modeling using linguistic factors. In: Proceedings of the 7th conference on speech technology and human—computer dialogue, (SpeD), Cluj-Napoca, Romania, pp 1–6. https://doi.org/10.1109/sped.2013.6682649
13. Alumae Z (2006) Sentence-adapted factored language model for transcribing Estonian speech. In: Proceedings of the IEEE international conference on acoustics speech and signal processing (ICASSP), Toulouse, France, pp 429–432. https://doi.org/10.1109/icassp.2006.1660049
14. Hirsimaki T, Pylkkonen J, Kurimo M (2009) Importance of high-order n-gram models in morph-based speech recognition. IEEE Trans Audio Speech Lang Process 17(4):724–732. https://doi.org/10.1109/tasl.2008.2012323
15. Mousa A, Shaik M, Schlüter R, Ney H (2011) Morpheme based factored language models for German LVCSR. In: Proceedings of the annual conference of the international speech communication association, INTERSPEECH, Florence, pp 1445–1448
16. Choueiter G, Povey D, Chen, S.F., Zweig G (2006) Morpheme based factored language models for Arabic LVCSR. In: Proceedings of the IEEE international conference on acoustics speech and signal processing, Toulouse, France, pp 1053–1056. https://doi.org/10.1109/icassp.2006.1660205
17. Adel H, Vu NT, Kirchhoff K, Telaar D, Schultz T (2015) Syntactic and semantic features for code-switching factored language models. IEEE/ACM Trans Audio Speech Lang Process. 23 (3):431–440. https://doi.org/10.1109/taslp.2015.2389622
18. Adel H, Kirchhoff K, Telaar D, Thang V, Schlippe T, Schultz T (2014) Features for factored language models for code-switching speech. In: Proceedings of the SLTU-2014, St. Petersburg, Russia, pp 32–38
19. Kirchhoff K, Vergyri D, Bilmes J, Duh K, Stolcke A (2006) Morphology-based language modeling for conversational Arabic speech recognition. Comput Speech Lang 20(4):589–608
20. Ganji S, Sinha R (2018) A novel approach for effective recognition of the code-switched data on monolingual language model. In: Proceedings of the Interspeech 2018, Hyderabad, pp 1953–1957. https://doi.org/10.21437/interspeech.2018-1259
21. Gregor D. Zdravko K (2017) Context-dependent factored language models. EURASIP J Audio Speech Music Process 2017, Article No.: 104 https://doi.org/10.1186/s13636-017-0104-6
22. Koehn P, Hoang H (2007) Factored Translation Models. In: Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), association for computational linguistics, Prague, Czech Republic, pp 868–876. https://www.aclweb.org/anthology/D07-1091.pdf
23. Nair J, Krishnan A, Deetha K (2016) An efficient English to Hindi machine translation system using hybrid mechanism. In: Proceedings of the international conference on advances in computing, communications and informatics (ICACCI), Jaipur, India. https://doi.org/10.1109/icacci.2016.7732363
24. Ramanathan A, Choudhary H, Ghosh A, Bhattacharyya P (2009) Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, Suntec, Singapore, pp 800–808. https://www.aclweb.org/anthology/P09-1090.pdf
25. Sreelekha S, Bhattacharyya P (2017) Role of morphology injection in SMT: a case study from Indian language perspective. ACM Trans Asian Low-Resour Lang Inf Process 17(1):1. https://doi.org/10.1145/3129208
26. Patel R, Pimpale P, Sasikumar M (2018) Machine translation in Indian languages: challenges and resolution. J Intell Syst 1:5. https://doi.org/10.1515/jisys-2018-0014
27. Jaya K, Gupta D (2016) Exploration of corpus augmentation approach for English–Hindi bidirectional statistical machine translation system. Int J Electrical Comput Eng (IJECE) 6 (3):1059–1071. https://doi.org/10.11591/ijece.v6i3.8904
28. Kumar A, Dhanalakshmi M, Soman K, Rajendran S (2014) Factored statistical machine translation system for English to Tamil language. Pertanika J Soc Sci Humanit 22(4):1045–1061
29. Dungarwal P, Chatterjee R, Mishra A, Kunchukuttan A, Shah R, Bhattacharyya P (2014) The IIT Bombay Hindi to English translation system at WMT 2014. In: Proceedings of the ninth workshop on statistical machine translation, association for computational linguistics, Baltimore, Maryland USA, pp 90–96. https://www.aclweb.org/anthology/W14–3308.pdf, https://doi.org/10.3115/v1/w14-3308
30. Sachdeva K, Srivastava R, Jain S, Sharma D (2014) Hindi to English machine translation: using effective selection in multi-model SMT. In: Proceedings of the ninth international conference on language resources and evaluation (LREC'14) Reykjavik, Iceland, European Language Resources Association (ELRA), pp 1807–1811. http://www.lrec-conf.org/proceedings/lrec2014/pdf/682_Paper.pdf
31. García-Martínez M, Barrault L, Bougares F (2017) Neural machine translation by generating multiple linguistic factors. In: Proceedings of the conference on statistical language and speech processing (SLSP), pp 21–31. https://doi.org/10.1007/978-3-319-68456-7_2
32. Wilken P, Matusov E (2019) Novel applications of factored neural machine translation. https://arxiv.org/pdf/1910.03912.pdf
33. Hokamp C (2017) Ensembling factored neural machine translation models for automatic post-editing and quality estimation. In: Proceedings of the conference on machine translation, Association for Computational Linguistics, vol. 2: shared task papers, Copenhagen, Denmark, pp 647–654. https://www.aclweb.org/anthology/W17-4775, https://doi.org/10.18653/v1/w17-4775
34. Brants T, Popat A, Xu P, Och F. Dean J (2007) Large language models in machine translation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, Association for Computational Linguistics, Prague, pp 858–867
35. Katz SM (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans Acoust Speech Signal Process 35(3):400–401. https://doi.org/10.1109/TASSP.1987.1165125