Generation of Missing Words in Assamese text using N-gram based Model

M. P. Bhuyan
Department of Information Technology, Gauhati University, Guwahati, India
E-mail: mpratim250@gmail.com

S. K. Sarma
Department of Information Technology, Gauhati University, Guwahati, India
E-mail: sks001@gmail.com

Abstract. It is very common to miss certain words when writing while listening to others. A similar problem can arise when typing on the computer. The automatic generation of missed words shall very much helpful for users by suggesting the required words. In this research work, missed words of the Assamese sentences are generated, at present, there is no such tool/method exists which can provide or generate the missed words in an Assamese sentence. N-gram based models like bigram and trigram are used to generate missed words. Using the bigram and trigram models a rank is calculated for each possible suggested words and the suggested word list is sorted according to this rank value in decreasing order. Finally, these suggested words can be used to fill the place of the missed Assamese words. Different levels of experiments are carried out and the present proposed system can correct the missed words at an accuracy ranging from 58% to 66%. The proposed model can precisely generate accurate five relevant suggestions for a sentence containing an average of six words with two missed words separated by three words.

1. Introduction
The Assamese language is known as Asomiya, the language is spoken in Assam, Arunachal Pradesh, Meghalaya. Assamese is an official language of Assam. Almost 20 million speakers are found in this language. People living in the Brahmaputra valley mostly speak Assamese. Assamese is an Indo-Aryan language which was evolved in the 7th century and the roots of the Assamese language are seen in the Sanskrit language. But, the literature of the Assamese language comes in the 14th century. History says that from Magadhi Prakrit Assamese and other related languages like Bengali, Oriya is originated. Magadhi Prakrit later evolved into the Eastern Indo-Aryan languages Bihari, Bengali, and Assamese languages which is also the source of Aprabhramsa dialects in Brahmaputra valley. Assamese literature is also very ancient and rich, Hema Saraswati was the antediluvian and among the earliest Assamese writer who wrote the poem, Prahlada Charita one of the earliest poetic works in the Assamese language which was written at the end of the 13th century. Madhava Kandali an Assamese poet was another very well known figure in Assamese literature who wrote the famous epic Ramayana in Assamese. Assamese script is evolved from the Kamarupi script. The ancient Assamese script was found
in Umachal (Nagajari Khanikargaon) rock inscription which was earlier than 7th century. Three different styles of Assamese script were found in the 17th century they were bamiiniya, kaitheli and garhgamma, the present Assamese script has some similarity with the Bengali script.

From the above discussion, it is cleared that Assamese is an old language and rich in literature. In this research, Assamese language is taken as the resource language and the language is not explored very much computationally. This research work is focusing on the missed words in an Assamese sentence. Bigram and trigram based models are used to generate the suggestion for the missed words. At present there is no such work on Assamese language to generate suggestions for missed words, only non-word error detection, and correction techniques are seen in earlier research. Let us have a closer look at the syntax of the Assamese language and understand the missed word problem clearly, consider the following example:

Assamese_text1="আেহামসকলৰ সুদীঘা অসম শাসনৰ মূলেত আিছল ততওঁেলাকৰ উদাৰ নীিত। ততওঁেলােক সমসামিয়ক যুগে জাতিতেন পথতাক গণব নকিৰিছিল। জংমজংম আেহাম বাজীত বসবাস কবা তৈয়াম আক জনজাতীয় লোকসকলকে ততওঁেলোকে আেহাম সমাজভু কবিল।"

The above sentences are correct in Assamese. Now, let us delete some words and generate some missed word errors.

Assamese_text2="আেহামসকলৰ সুদীঘা অসম শাসনৰ মূলেত আিছল ততওঁেলাকৰ _______ নীিত। ততওঁেলােক সমসামিয়ক জাতিতেন পথতাক গণব নকিৰিছিল। জংমজংম আেহাম বাজীত _______ কবা তৈয়াম আক জনজাতীয় লোকসকলকে ততওঁেলোকে আেহাম _______ কবিল।"

If Assamese_text1 and Assamese_text2 are compared the missing words can be identified which are underlined in the following Assamese_text3.

Assamese_text3="আহোমসকল সুদীঘা অসম শাসন মূলতে আছিল ততওঁেলকে নীতি। ততওঁেলাকে সমসামিয়ক জাতিতেন পথতাক গণব নকিৰিছিল। জংমজংম আেহাম বাজীত কবা তৈয়াম আক জনজাতীয় লোকসকলকে ততওঁেলোকে আেহাম সমাজভু কবিল।"

This is not an uncommon problem which arises when someone types long document or book, there may be the chance of missing some words due to exhaustive typing. If such types of errors can be rectified by making the editing software a little intelligent then many users shall get benefited and need not bother about if such mistakes are done at some point. The presence of such assisting software will increase the confidence of the user as a result throughput of the user will be catalyzed and also help the new learners, differently-abled people. From this discussion, it is straightforward that troubleshooting the missing-word problem is valid in the present scenario. The rest of the paper is organized as follows: Section 2 describes related works in this research work, Section 3 describes the Proposed Model, Section 4 analyzes and discuss the experimental results, Section 5 concludes the current work and proposes the future direction of this research work.

2. Related Works

Mridha et al. [1] have described the method of detection and correction of missing words in Bengali sentences. They have proposed a method to detect and generate suggestions for the missed words with an accuracy of 82.82%. Their method was basically n-gram based, they have used two corpora one was bigram, which was used to detect the errors, and the other was trigram which was used to correct the errors. Finally, they have evaluated their method on six different corpora. In their proposed model, they have generated bigrams for a sentence. These bigrams were searched one by one in the bigram corpus and if the bigrams were found in the bigrams corpus then there was no word missing error. On the other hand, if any bigram was not found in the bigram corpus then the system would declare an error and generated the suggestion using the trigram corpus. In the trigram corpus, all the trigrams were returned whose first and the last word was the first and last word of the bigram which was not found in the bigram corpus, all such trigrams were sorted according to their frequency of occurrences and their middle words are displayed as a suggestion to overcome the missed word error.

Salah et al. [2] had proposed a method to detect missed text of Optical Character Recognition (OCR) system, OCR systems are very complex design composed of many subsystems dedicated to analyze and
recognize elements of a page. Missed text is one of the frequent errors in OCR, occurrences of such errors effects the digital content libraries. They have detected the missed text of OCR for French National Library collections and the accuracy of their method was 84.15%.

Samanta and Chaudhuri [3] have discussed a technique to detect and correct real-word errors using bigram and trigram. They have calculated the score of the bigram generated by the erroneous word and its left and right neighbors and the trigram of these three words. A single character error was considered, i.e. if the word is erroneous then the correct word will belong to the real word list which is formed by a single character edit operation on the erroneous word. The real word list is ranked based on the score value of each word of the list. Using the score value decision was made on every word and suggestions were generated accordingly. They have designed a confusion set of each word by doing insertion, deletion, substitution, etc. and finally, a weighted score, by combining two bigrams and one trigram was calculated for each word of the confusion set. They have achieved satisfactory performance on a moderate size test set and comparable with those state of art real word error correction techniques.

Saharia and Konwar [4] designed one Assamese writing software named as LuitPad. They have focused on the two main typing options one is phonetic sound of words and the second one is the phonetic sound of the characters. LuitPad was enabled with the spell checking option with the help of an internal dictionary of size 60,000 words. Assamese characters and words are retrieved by using the English alphabet. LuitPad had two primary input modes: character mode and word mode. In character mode, one character was inputted at a time and word mode one word was inputted at a time. In character mode, the Assamese alphabets were mapped to Roman alphabets, this mapping was customizable by the user. Multiple users could design their profiles with their preferred way of pronunciation. In word mode, the whole Assamese word was entered by approximate matching of the phoneme. These phonemes might help the user to map different words to one word. LuitPad spell checker was another added feature included by Saharia and Konwar, this spell checker was based on the dictionary words, if the word entered by the user was not found then the word was underlined with a red line. The user might ignore the spelling error by clicking at the ignore tab or might use suggestions by right-clicking at the erroneous word. Spell checking feature was basically by calculating the Levenshtein distance between two words. Finally, they have tested 14 different users to check the performance of the system, words per minute recorded were 8 to 22, spelling checking accuracy was 93%.

Bhuyan and Sarma [5] proposed a model for automatic word formation, correction, and termination of Assamese words. They mathematically evaluate their model using the various n-gram models and experimentally found the sentence probability for unigram, bigram, trigram, and quadrigram model. Besides, the experimental results showed that the trigram and quadrigram models' performance almost the same. Bhuyan and Sarma [6] have given an N-gram based model to predict the Assamese words. They used unigram, bigram, trigram, and quadrigram model, and experiments are performed to check the Keystrokes Saving and Keystrokes Until Completion. They were able to achieve the state-of-the-art performance of their system for the pre-configured and the user's input dataset. Bhuyan and Sarma [7] have performed another experiment on the accuracy of Assamese word prediction against the prediction list length. If the length of the prediction list is more then the list will have more words and there is a more chance of appearing the user's word in the predicted words. They have carried out two experiments, one for in-domain data, and the second one was for mixed domain data. Both the experiment were tested from the prediction list length one to ten for the first four n-gram models. They have concluded that the prediction list should not be very large although the increase of prediction list results in the increase in accuracy of prediction, yet the user would be overloaded with the number of options which one to select that went to a time-consuming task. Finally, they have concluded that the prediction list length should be either five or six which was adopted in most of such predictive models and this length value was experimentally proved.

Bhuyan and Sarma [8] designed a statistical model for the detection and correction of Assamese words. They have used character-level n-gram models, they have collected fourteen pairs of Assamese symbols/alphabets which are similar phonetically and a score based system was designed to assign a score
value for each of the words they tested and depending on a predefined threshold value the word was decided as correct or wrong. The model was able to achieve an accuracy of around 80%. Bhuyan and Sarma [9] proposed a model to enhance automatic Assamese word prediction using higher-order n-gram models for ambiguous words. In this research work, they have used up to the 6-gram model. They were able to increase the accuracy of 6% and the failure rate by 3% compared with the traditional Assamese word prediction model. Moreover, this enhanced model worked like an add-on to the traditional predictive model so if the enhanced model failed then the traditional predictive model started to play. Consequently, the enhanced model tried to increase the performance if failed then the performance of the system would not decrease. Bhuyan et al. [10] have designed a model using the bigram and trigram model to categorize the Assamese words based on their context. They designed a score based system and according to the predefined threshold value the words are clustered in the various clusters. After doing the clustering the clusters were observed manually by the linguistics experts and finally, they were able to form 733 clusters with the clustering accuracy around 60%. Bhuyan et al. [11] have designed a statistical model to check the correctness of the Assamese sentences. They have used the linear interpolation of the n-gram models and using the trigram probabilities of the words the sentence probability is evaluated. If the sentence probability is higher than the predefined threshold value then the sentence is valid otherwise invalid. Experiments were carried out for the correct, incorrect, and mixture of correct and incorrect sentences, the F1-score obtained in the experiment was around 60%.

Bhuyan et al. [12] have designed a model for information retrieval in the Assamese language using WordNet and Assamese Wikipedia. The IR system is consists of three phases viz. Query formation, query expansion using WordNet, and retrieving information from Assamese Wikipedia. The performance of the system is around 60%. Deka et al. [13] have studied the various works and progress in the Assamese language from a natural language processing perspective and they have mentioned the processing-importance of the local languages in the computational field. Kalita et al. [14] have designed an event detection model using Conditional Random Filed, they have designed a dataset to detect the events in Assamese the accuracy of the model was 79%. Rahman et al. [15] have proposed a model for converting the printed Assamese numerals to the corresponding utterance using the image processing technique. The numerals are pronounced using Google sounds. The accuracy of recognition was 70%. Sarma et al. [16] proposed one method using an Artificial Neural Network to recognize the spoken digits by the North-Eastern people, the system was trained using the voice samples of Assamese people. The digit was displayed after successful recognition. The accuracy of the system was 82%.

Welleck et al. [17] have proposed a method for generating words in non-monotonic orders, the model learns the good order and was able to generate the words at any arbitrary position. Iqbal and Qureshi [18] did a study on various deep learning methods for automatic text generation. In this work, they have summarized deep learning text generation models. They have discussed the word2vec model, Long Short Term Memory(LSTM), Bi-directional RNN, Variational Auto-encoder, Generative Adversarial Networks, etc. They have mentioned that the availability of text data and deep learning model make the NLP problems easier. Bahdanau et al. [19] have presented an approach for training a neural network using the actor-critic method for supervised learning problems to get structure output. The actor-critic model can fit the training data in a faster way. Pianta and Tovena [20] presented a hybrid approach for automatic text generation. They mixed the template-based system with the Natural Language Generation(NLG) system. The NLG model depends on the word, morphemes, sentence, syntactic and semantic features of a language. Pawade et al. [21] have designed a model termed as story scrambler using Recurrent Neural Network(RNN). They have tried to generate new stories by feeding the RNN with some series of input stories. They have studied various automatic text generation techniques and summarized them based on two parameters one was the purpose and the second one was the methodology used. They were able to achieve an accuracy of 63% which was evaluated manually. Spiccia et al. [22] have proposed a new method for automatic sentence completion which was based on Latent Semantic Analysis with word-word frequency matrix. Experimental results showed that the proposed method was able to improve the results than the other non neural network based language model. The model was able to generate the missed
words by assigning a score value.

From the above-related study, it is seen that natural language processing is an emerging area in the field of Artificial Intelligence. The working of the local languages is under progress and the various works for the development of the Assamese language are also going on. So, the generation of the missed word in Assamese sentences is a valid work and will advance the Assamese language development and processing.

3. Proposed Model
This section describes the method used in this research work. The following subsections describe the corpus used in the experiments, algorithms for detecting and generating suggestions, etc.

3.1. Corpus preparation
A corpus of 43,000 sentences containing a total of 250,000 words is prepared. The corpus contains text from Assamese newspapers, books, magazines, etc., two testing corpora with a mixed domain data (i.e. training data and non-training data) are designed by deleting one word from each sentence of the original corpus and the other containing by deleting two words from the original corpus. Bigrams and Trigrams of the corpus words are calculated and stored along with their frequency of occurrences in BiCorpus and TriCorpus respectively. For the detection and generation of the missed words left bigram, right bigram, and trigram all these three are used. For the missed word $W_i^m$ following bigrams and trigram are found:

Left Bigram: $W_{i-1}^j W_{mj}$
Right Bigram: $W_{mk}^i W_{i+1}$
Trigram: $W_{mp}^{i-1} W_{m}^i W_{i+1}$

$Lrb = lengthOfRight(bigramListOf(W_{i-1}^j))$
$Llb = lengthOfLeft(bigramListOf(W_{i+1}^j))$
$Lt = lengthOfTrigram(trigramListOf(W_{i-1}^j W_{mp}^{i-1} W_{i+1}^j))$

where $0 \leq j \leq Lrb, 0 \leq k \leq Llb, 0 \leq p \leq Lt$

3.2. N-gram probability Estimation
Markov’s chain rule can be used to calculate the probability of a sentence. But, Markov’s assumption says that the probability of future events depends only on the few previous events. For example, sentence probability can be calculated in the bigram language model as follows:

$$p(W_1, W_2, W_3, \ldots, W_n) = p(W_1) \times p(W_2|W_1) \times p(W_3|W_2) \ldots p(W_n|W_{n-1}) \times p(W_n)$$  \hspace{1cm} (1)

In the present proposed model sentence probability is not calculated. Considering the Markov assumption, it is assumed that the occurrence of any event depends only on the immediate next and previous events. Bigram and trigram probabilities are calculated as follows:

$$p_1(W_{mj}^i | W_{i-1}^j) = \frac{\text{count}(W_{mj}^{i-1} W_{mp}^{i-1})}{\sum_{s=0}^{Lrb} \text{count}(W_{ms}^{i-1} W_{mp}^{i-1})}$$  \hspace{1cm} (2)

$$p_2(W_{mk}^i | W_{i+1}^j) = \frac{\text{count}(W_{mk}^{i-1} W_{mp}^{i+1})}{\sum_{s=0}^{Llb} \text{count}(W_{ms}^{i} W_{mp}^{i+1})}$$  \hspace{1cm} (3)

$$p_3(W_{mp}^i | W_{i-1}^j W_{i+1}^j) = \frac{\text{count}(W_{mp}^{i-1} W_{mp}^{i+1})}{\sum_{s=0}^{Lt} \text{count}(W_{ms}^{i-1} W_{mp}^{i+1})}$$  \hspace{1cm} (4)

Using equation (2), probability $p_1$ for each bigram of $W_{mj}^i$ is calculated whose left bigram is $W_{i-1}^j$. The denominator in the equation (2) represents the sum of all bigrams whose left bigram is $W_{i-1}^j$. Bigram and trigram count can be obtained from the BiCorpus and TriCorpus respectively. Similarly, $p_2$ is calculated for right bigram count. For every element of the left and right bigram list probabilities are
calculated, so that following conditions are satisfied:

$$\sum_{s=0}^{L_{lb}^y} p_1(W_{mx}^i|W^{i-1}) = 1$$

$$\sum_{s=0}^{L_{lb}^y} p_2(W_{mx}^i|W^{i+1}) = 1$$

In equation (4) trigram counts are used from the TriCorpus. In equation (4) denominator represents the sum of all trigrams containing $W_i^1$ in the left and $W_i^{i+1}$ in the right of the trigram. For every middle words of $W_i^1$ and $W_i^{i+1}$ are calculated so that

$$\sum_{s=0}^{L_{lt}^y} p_3(W_{mx}^i|W^{i-1}W^{i+1}) = 1$$

Probabilities values $p_1$, $p_2$, and $p_3$ which are obtained from equation (2), (3), and (4) are combined to generate a Rank value for $W_i^{mx}$ which may be placed to fill the gap of the missed word.

$$\text{Rank}(W_i^{mx}) = p_1(W_{mx}^i|W^{i-1}) + p_2(W_{mx}^i|W^{i+1}) + p_3(W_{mx}^i|W^{i-1}W^{i+1})$$  (5)

In the above equation (5), it is cleared that $0 \leq \text{Rank}(W_i^{mx}) \leq 3$. Simple addition will not produce a good sense because the higher-order n-grams give more contextual information than the lower-order n-grams. By using the principle of interpolation weighted combination of Rank values are calculated as shown below:

$$\text{Rank}(W_i^{mx}) = \alpha_1 p_1(W_{mx}^i|W^{i-1}) + \alpha_2 p_2(W_{mx}^i|W^{i+1}) + \alpha_3 p_3(W_{mx}^i|W^{i-1}W^{i+1})$$  (6)

$\alpha_1$, $\alpha_2$, and $\alpha_3$ are found using trial and error method in the training set and the best results are found if $\alpha_1 = 0.20$, $\alpha_2 = 0.20$, and $\alpha_3 = 0.60$. so that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Finally, the $\text{Rank}(W_i^{mx})$ will satisfy $0 \leq \text{Rank}(W_i^{mx}) \leq 1$.

3.3. Missing word Detection

Following Algorithm 1 describes the detection of missing words in a sentence.

sent = sentence to be tested

Bigram list of sent, bgList = $W_1W_2, W_2W_3, W_3W_4 \ldots W_n-1W_n$

**Algorithm 1** Missing word Detection

1. for each bigram bg ∈ bgList do
2. if bg ∈ BiCorpus then
3. Declare no word is missed and continue to the next
4. else
5. Declare a word may be missed and Generate suggestion
6. end if
7. end for

3.4. Suggestion Generation

If the bigram is not found in the BiCorpus following Algorithm 2 is used to generate suggestions for the missed word.

4. Result and Discussion

In this section, the experimental results are tested and analyzed. Experiments are performed for different size of dataset the datasets are containing a mixture of in-domain and out-domain data, a mixed dataset is used to know the actual performance of the system. Because, if the system is tested only for the in-domain data then the system will be biased and if is tested for out-domain data then the system will show abrupt behavior, therefore, a mixture of both in-domain and out-domain data is chosen for the experiments. The experiments and their results are shown below:

- Experiment-1: Deleting a single word from the sentence
- Experiment-2: Deleting two words from the sentence
Algorithm 2 Suggestion Generation

1: if $W_iW_{i+1} \notin bgList$ then
2: \hspace{1em} $W_{mx}^i$ may be the missed word
3: \hspace{1em} Extract the left bigram list of $W_{mx}^i$, bigram list $Lb$ of $W_i$ from the bigram corpus.
4: \hspace{1em} Calculate $p_1$ for each $W_{mx}^i \in Lb$
5: \hspace{1em} Extract the right bigram list of $W_{mx}^i$, bigram list $Rb$ of $W_{i+1}$ from the bigram corpus.
6: \hspace{1em} Calculate $p_2$ for each $W_{mx}^i \in Rb$
7: \hspace{1em} Extract the trigram list containing $W_{mx}^i$ as middle word, trigram list $TL$ of $W_i$ and $W_{i+1}$ as a first and third word from the trigram corpus.
8: \hspace{1em} Calculate $p_3$ for each $W_{mx}^i \in TL$
9: end if
10: $suggList = \phi$
11: for each $W_{mx}^i, W_{mx}^i \in Lb$ or $W_{mx}^i \in Rb$ or $W_{mx}^i \in TL$ do
12: \hspace{1em} Create a list suggestedWord($W_{mx}^i, Rank(W_{mx}^i)$)
13: \hspace{1em} $suggList = suggList + suggestedWord(W_{mx}^i, Rank(W_{mx}^i))$
14: end for
15: sortDecreasingRank($suggList$)
16: display($suggList$)

In both the two experiments different tests are carried out on the number of sentences ranging from 1000 to 5000. Let us consider the sentence $S$= আহোমসকল সুদীর্ঘ অসম শাসন মূলেত আছিল তেল্লোক উদার নীতি। The fifth word “মূলেত” is deleted from the sentence and the suggestions along with the score of the related words are generated by the model as shown in Table 1.

| Sl. No. | Words($W_i$) | Rank(decreasing order) |
|--------|--------------|------------------------|
| 1      | মূলেত        | 0.0040                 |
| 2      | আবারুরপ্ত  | 0.0038                 |
| 3      | পর্ববর্তীকে   | 0.0029                 |
| 4      | মাঝত        | 0.0022                 |
| 5      | লগত        | 0.0018                 |

Table 2. Experiment-1: Deleting a single word from the sentence

| Sl. No. | #sentences | Precision | Recall | Accuracy |
|---------|------------|-----------|--------|----------|
| 1       | 1000       | 68.10%    | 75.04% | 65.80%   |
| 2       | 2000       | 65.15%    | 64.69% | 60.50%   |
| 3       | 3000       | 67.55%    | 72.52% | 65.60%   |
| 4       | 4000       | 68.74%    | 77.19% | 65.47%   |
| 5       | 5000       | 68.83%    | 67.57% | 66.96%   |
Table 3. Experiment-2: Deleting two words from the sentence

| Sl. No. | #sentences | Precision | Recall   | Accuracy  |
|---------|------------|-----------|----------|-----------|
| 1       | 1000       | 63.00%    | 73.56%   | 62.60%    |
| 2       | 2000       | 62.31%    | 63.27%   | 58.75%    |
| 3       | 3000       | 67.49%    | 72.51%   | 65.57%    |
| 4       | 4000       | 67.94%    | 76.53%   | 64.90%    |
| 5       | 5000       | 67.98%    | 66.99%   | 66.40%    |

Table 2 shows the results for the sentences containing one missed word with an accuracy ranging from 60% to 66% and Table 3 shows the results for the sentences containing two missed words with an accuracy ranging from 58% to 66%. The experimental results are nearly stable with the increasing number of sentences.

5. Conclusion and Future Work
The missed word generation in the Assamese language is not explored until now. This is the first research work in the Assamese language. This model will help the people while typing Assamese text at a rapid rate by automatically filling the missed words. The performance of the system satisfactory and it is possible to achieve an average level of accuracy in this work. But, the model is consistent with the increased number input size which is a remarkable result obtained from the experiments. In the future, the model can be tested by deleting more words from a sentence. Secondly, the failure rate of the system is due to the unavailability of the data in the system, the performance of the model can be improved by increasing the corpus size. The introduction of a deep learning model can be another scope of future study and application of this model to other less explored languages can also be a future direction of study.

References
[1] Mridha M. F., M. M. Rana, M. A. Hamid, M. E. A. Khan, M. M. Ahmed and M. T. Sultan 2019 An Approach for Detection and Correction of Missing Word in Bengali Sentence International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox’s Bazar, Bangladesh, pp. 1-4. doi: 10.1109/ECACE.2019.8679416.
[2] Salah A. B., N. Ragot, T. Paquet and T. Paquet 2013 Adaptive detection of missed text areas in OCR outputs: application to the automatic assessment of OCR quality in mass digitization projects SPIE Document Recognition and Retrieval XX, Feb 2013, SAN FRANCISCO, United States. 8658, pp.110-122.
[3] Samanta, P., and B. B. Chaudhuri 2013 A simple real-word error detection and correction using local word bigram and trigram In Proceedings of 25th Conference on Comput. Linguist. Speech Process (ROICING’13) pp. 211-220.
[4] Saharia, N., and K.M Konwar 2012 LuitPad: A fully Unicode compatible Assamese writing software Proceedings of the Second Workshop on Advances in Text Input Methods (WTIM 2), COLING, Mumbai, December 2012, pp. 79-88.
[5] Bhuyan M. P. and Sarma S.K. 2018 Automatic Formation, Termination & Correction of Assamese word using Predictive & Syntactic NLP Proceedings of the International Conference on Communication and Electronics Systems (ICCES 2018) DVD Part Number:CFP18AWO-DVD; ISBN:978-1-5386-4764-6, pp. 548-552.
[6] Bhuyan M. P. and Sarma S. K. 2019 An N-gram based model for predicting of word-formation in Assamese language Journal of Information and Optimization Sciences, Vol.40(2), pp. 427-440, DOI: 10.1080/02522667.2019.1580883
[7] Bhuyan M. P. and Sarma S. K. 2019 Effects of Prediction-Length on Accuracy in Automatic Assamese word prediction 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1-4.
[8] Bhuyan M. P. and Sarma S. K. 2019 A statistical model for automatic Error Detection and Correction of
[9] Bhuyan M. P. and Sarma S. K. 2019 A Higher-Order N-gram Model to enhance automatic Word Prediction for Assamese sentences containing ambiguous Words International Journal of Engineering and Advanced Technology, Vol.8(6) pp. 2921-2926, DOI: 10.35940/ijeat.F8706.088619.

[10] Bhuyan M. P., Sarma S. K., and Sarma P. 2020 Context-based Clustering of Assamese words using n-gram model Second International Conference on Advances in Electrical and Computer Technologies 2020 (ICAECT 2020), Springer (in press)

[11] Bhuyan M. P., Sarma S. K., and Rahman M. 2020 Natural Language Processing based Stochastic Model to Check the Correctness of Assamese Sentences 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 1179-1182, doi: 10.1109/ICCES48766.2020.9138067.

[12] Bhuyan M. P., Purkayastha R., Sarma S. K., Sarmah S., Sarma P., and Deka V. 2019 Information Retrieval in Assamese using Wordnet & Assamese Wikipedia International Journal of Scientific & Technology research, Vol.8(9), pp. 1887-1891.

[13] Deka R.R., Kalita S., Bhuyan M.P., and Sarma S.K. 2020 A Study of Various Natural Language Processing Works for Assamese Language In: Dawn S., Balas V., Esposito A., Gope S. (eds) Intelligent Techniques and Applications in Science and Technology. ICIMSAT 2019 Learning and Analytics in Intelligent Systems, Vol. 12 Springer, Cham, pp. 128-136.

[14] Kalita, S., Sarma, S.K., Bhuyan, M.P., Deka, V. 2020 Event Detection in Assamese Text Using Conditional Random Field Journal of Advanced Research in Dynamical and Control Systems, pp. 1370-1375, DOI:10.5373/JARDCS/V12SP4/20201615

[15] Rahman M., Sarma P., Bhuyan M. P., Das A., and Dutta D. 2019 Image to Speech Synthesizer with Reference to Assamese Numerals International Journal of Innovative Technology and Exploring Engineering, Vol.9(1), pp. 900-905, DOI: 10.35940/iijte.e.A4435.119119.

[16] Sarma P., Sarmah S., Bhuyan M. P., Hore K. K., and Das P. P. 2019 Automatic Spoken Digit Recognition Using Artificial Neural Network International Journal of Scientific & Technology Research, Vol.8(12), pp. 1400-1404.

[17] Welleck S., Bratley K., Daume H., and Cho K. 2019 Non-Monotonic Sequential Text Generation. arXiv preprint arXiv:1902.02192v3.

[18] Iqbal T. and Qureshi S. 2020 The survey: Text generation models in deep learning Journal of King Saud University – Computer and Information Sciences https://doi.org/10.1016/j.jksuci.2020.04.001

[19] Bahdanau D., Brakel P., Xu K., Goyal A., Lowe R., Pineau J., Courville A., Bengio Y. 2016 An actor-critic algorithm for sequence prediction arXiv preprint arXiv:1607.07086.

[20] Pianta E. and Tovenia L. 1999 Mixing representation levels: The hyrbid approach to automatic text generation Proceedings of the AISB’99 Workshop on Reference Architectures and Data Standards, arXiv:cs/9907026v1 [cs.CL] University of Edinburgh.

[21] Pawade D., Sakhapara A., Jain M., Jain N., Gada K. 2018 Story scrambler-automatic text generation using word level RNN-LSTM International Journal of Information Technology and Computer Science(IJTCS), Vol.10(6), pp.44-53, DOI: 10.5815/ijtcs.2018.06.05

[22] Spiccia C., Augello A., Pilato G, and G. Vassallo 2015 A word prediction methodology for automatic sentence completion Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, CA, pp. 240-243. doi: 10.1109/ICOSC.2015.7050813