International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

N-gram based Machine Translation for English-Assamese: Two Languages with High Syntactical Dissimilarity

Zakir Hussain, Malaya Dutta Borah, Abdul Hannan

Abstract: ***To bridge the language constraint of the people residing in northeastern region of India, machine translation system is a necessity. Large number of people in this region cannot access many services due to the language incomprehensibility. Among several languages spoken, Assamese is one of the major languages used in northeast India. Machine translation for Assamese language is limited compared to other languages. As a result, large number of people using Assamese language cannot avail lots of benefits associated with it. This paper has focused on the development of the English to Assamese translation system using n-gram model. The n-gram model works very well with the language pair having high dissimilarity in syntax compared to other models. The value of n has a very big role in the quality and efficiency of the system. Bilingual Evaluation Understudy (BLEU) score differs significantly with the change of the n-gram. This model uses tuples to reduce the consumption of excess memory and to accelerate the translation process. Parallel corpus has been used for training the n-gram based decoder called MARIE. The number of translation units extracted using n-gram model is much less than the translation units extracted using phrase based model. This has a high impact on system efficiency.***

Keywords: Statistical Machine Translation, N-gram, MARIE, English-Assamese Translation, Tuple Extraction

I. INTRODUCTION

Machine translation is a subfield of computer linguistics that deals with the translation carried out by the computers. In the context of text translation, user inputs texts of one standard language and the system translates the same into the texts of another standard language. During this process, the system needs to follow some rules specifically applicable to the target language. Since the syntax of different languages is different, the system should understand the rules of different languages. The position of noun, verb and object of different languages are different; hence, it should be taken care of such that the sentences are translated with proper meaning without violation of grammatical rules. The quality of the translation is measured on the basis of some factors like handling the linguistic typology, translation of the idioms and how the anomalies are isolated. Now-a-days, machine translation has got much importance in different fields. Earlier, language was a very big barrier in sharing ideas and knowledge between diverse language speakers. Machine translation has attracted various domain of work due to its ability of bridging the gap created by that barrier. The significant demand for translation of electronic text on the internet, such as web pages, e-mail, social media, electronic chat, official document translation etc. is being noticed from past few years. To fulfill the need of translation, the following well defined approaches are referred by the research community in the area of machine translation (Please refer to Fig.1 for pictorial view of the approaches).

- **Direct approach**: The direct approach is the simplest one. Translation is done word-by-word basis. In this approach, no linguistic analysis of the source sentence is taken into consideration for producing a target sentence. Now-a-days this approach has been abandoned even in the corpus-based framework.
- **Rule based approach**: This approach is based on the rules generated by the human experts [11], [17]. Different human experts may specify different rules for translation process. Hence, for different persons, the system will be of different configuration and of different efficiency. This approach can be subdivided into:
  - *Transfer approach*: The transfer approach has three phases. First one is the *analysis phase* where analysis of the source sentence is done to produce an abstract representation. Second one is the *transfer phase* where abstract representation of the first phase is being transferred into the equivalent representation in the target language. Third one is the *generation phase* where target sentence is generated from the intermediary representation.
  - *Interlingua approach*: This approach produces the target sentence based on the Interlingua representation created by thorough analysis of the syntax and semantics of the source sentence. This approach analyses the source sentence deeply. The advantage of Interlingua approach is that once the meaning of the source sentence is grasped, it can be articulated in any number of target languages.
- **Corpus based approach**: This type of system extracts the knowledge by analyzing the translation examples from parallel corpus. A parallel corpus is a collection of texts of more than one language, each of which is an exact translation of each other. The parallel corpus is developed by human experts. Here, the translation system can be developed as soon as the required technique is ready for a given pair of languages. A corpus-based approach generally follows direct or transfer approach. The Corpus based approach can be further

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Zakir Hussain*, Research Scholar, Department of CSE, NIT Silchar, Assam, India.
Malaya Dutta Borah, Assistant Professor, Department of CSE, NIT Silchar, Assam, India.
Abdul Hannan, Faculty, Department of IT, Gauhati University, Assam, India.

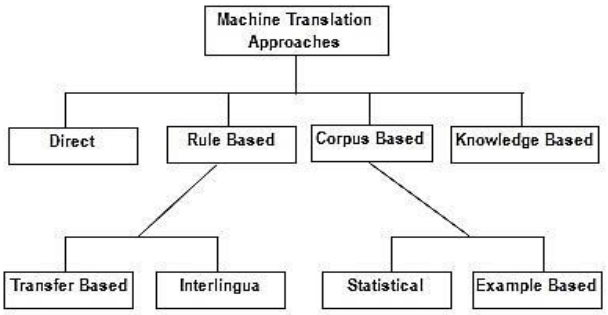

Fig. 1. Approaches to Machine Translation

subdivided into the following two categories:

- *Statistical*: It uses the examples from the parallel corpus for the training of the system. Statistical approach initially worked only on word-by-word fashion. But now-a-days, it aims to introduce linguistic analysis to some extent.
- *Example based*: It uses the examples from the parallel corpora. Translation is provided by choosing and combining the examples.

- **Knowledge based**: This approach requires *ontological knowledge* and *lexical knowledge*. Ontological knowledge means the description of the concepts and relationships that can exist for the agents and lexical knowledge means all the information that is known about the words and the relationships among them [5].

In recent research, rule based and corpus based approaches have got importance as these approaches have shown significant improvement in the quality of the translation. Due to the high syntactical dissimilarity between English and Assamese languages, the researchers have found the corpus based approach as the suitable one. To accomplish the translation, different models extract translation units in different way (please refer to section 3 for details). In the earlier works on English to Assamese translation, the researchers extracted the translation units as the phrases. That worked well but the following limitations have been found:

- The extracted translation units (phrases) are more in number and hence consuming more memory.
- Finding the desired translated words from more number of units resulted more time to translate.

To address these issues, this work proposes an *n-gram based approach* which will certainly improve the capability of the translation system in terms of the *memory consumption* as well as the *translation time*.

The focal points of this approach are:

- Reduction in the number of extracted translation units (tuples) without compromising the quality of translation
- Reduction in the translation time.

In this work we are focusing only on English to Assamese translation because we have found that for other languages, a significant level of works in the field of machine translation has already been carried out and the people across the globe is availing the benefits. But the work is lagging behind for Assamese language and hence a very big number of people cannot avail lots of benefits associated with it.

The remaining portion of this paper is arranged into 5 sections. Section 2 is about the related works, section 3 presents the proposed work, section 4 shows the experimental setup and section 5 is about the results and discussions, while section 6 concludes the paper.

II. RELATED STUDY

Machine translation system was constrained being used in early days and could be seen uniquely in some areas like military administrations, national and international legislative administrations. For translating Russian military scientific and technical documentations into English, the US Air Force introduced Systran in 1970 [21]. Later, the translation system got importance in different fields and many works have been carried out for various languages. Only a few of those have been mentioned here. Using n-gram model, Spanish-English, Arabic-English and Chinese-English translation system was developed in the TALP Research Center under Speech Processing Group in the Signal Theory and Communications department of Universitat Polit`ecnica de Catalunya. MARIE, the n-gram based machine translation decoder was also been developed there [5]. English to South Dravidian translation system was developed at Computational Engineering and Networking Department, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India [23]. English to Sanskrit translation system has also been developed using lexical parser at Information Technology Department, Bharati Vidyapeeth University College of Engineering, Pune- 43, Maharashtra, India [1]. Translation system for English to Hindi was developed at Thapar University [20]. Likewise many more can be mentioned for different languages other than Assamese language. It has been seen that many popular translation systems like Google Translate, Bing Translator etc. have not worked for the Assamese language (Fig.2). During past a few years, the development of English-Assamese translation system has come into the interest of research community. Using phrase based decoder, translation system for English-Assamese and vice versa has been developed at Gauhati University [3] as well as at Dibrugarh University [19]. Both the systems have been implemented using very small corpus and MOSES as the decoder. MOSES is a phrase based machine translation decoder. In Gauhati University, a work using rule based machine translation approach for English-Assamese has also been done [9]. Till date limited works have been carried out for English to Assamese translation using n-gram model.

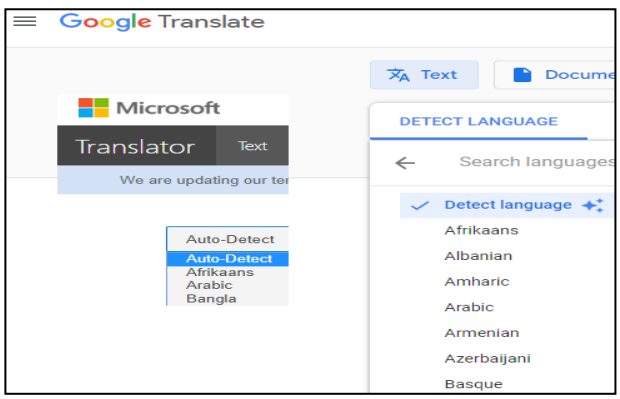

Fig. 2.Google and Bing Translator

Phrase based systems consume high amount of memory and takes more time to translate. Hence, n-gram based approach has been embraced to overcome these constraints to some extent.

III. PROPOSED WORK

Most of the existing translation systems for English to Assamese and vice versa have used the phrase-based approach. But for the language pair with high dissimilarity in the formation of sentences, phrase based system may produce inefficient translation. Hence, n-gram based system has been proposed.

Both n-gram based and phrase-based systems use word-by-word alignment of the corpora. After alignment, the *tuples* or *translation units* are extracted for n-gram based system and for phrase based system the *phrases* are extracted for performing the translation. The following paragraph shows how the tuples and phrases are extracted from the aligned sentence.

From the Fig. 3 we can extract the tuples (t1…tn) as follows:

t1: Dispur_is#দিছপুৰ

t2: the_capital_of_Assam #অসমৰ_ৰাজধানী

Any pair of the source sentence words and the target sentence words satisfying the following two constraints can be termed as phrase.

- Words should be consecutive along both the sides of the bilingual phrase, and
- Words on either side of the phrase should not be aligned to a word which is out of the phrase [5], [8].

So, keeping these two constraints in mind, the extracted phrases (p1….pn) from Fig. 3 are as follows:

p1: Dispur_is#দিছপুৰ

p2: Dispur_is_the_capital_of_Assam#দিছপুৰ_অসমৰ_ৰাজধানী

p3: the_capital#ৰাজধানী

p4: the_capital_of_Assam#অসমৰ_ৰাজধানী

p5: of_Assam#অসমৰ

From the above, it is observed that the number of extracted tuples is significantly less than that of extracted phrases. It is clear that the sentence pair can be segmented into multiple phrases ([p1+p4], [p2]). But, only single segmentation is possible when extracting tuples ([t1+t2]). This makes phrase based approach more robust than the n-gram approach. But the problem is that, phrase based approach consumes more memory than the n-gram based approach as the number of phrase is more than that of tuple. Another problem is the search time. N-gram based system accomplishes a similar level of translation in less time than that of phrase based system.

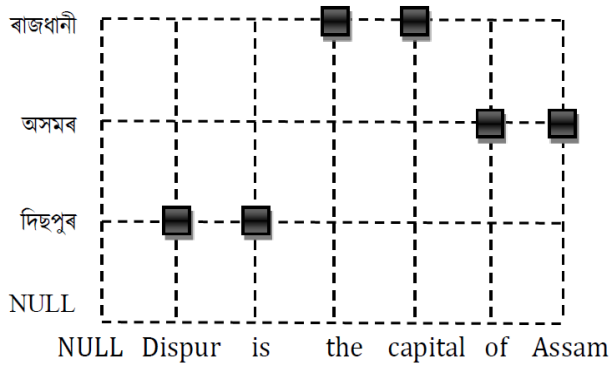

Fig. 3.Sentence alignment to show the extraction of tuple and phrase

A. *Background*

All statistical machine translation system essentially goes through three phases as follows:

- Language modeling: This phase involves the calculation of the probability of each word present in the target language corpus.
- Translation modeling: This phase takes care of the calculation of the probability of words present in the target language corpus given the probability of the words present in the source language corpus.
- Decoding: This phase involves the maximization of the probability to get the correct translation.

Phase-I: Language model

The language model provides the context of selecting the most prominent word to be placed after a sequence of words from many choices. The theoretical concept can be understood by this example. Consider an English sentence: *My phone is ringing*. One translated sentence in the Assamese language is মোৰ ফোন বাজি আছে. In the translation process, after placing মোৰ, the next word to be placed may be ফোন or বাজি or আছে. But during the probability calculation, probability of getting ফোন will be greater than বাজি and আছে. i.e. P(ফোন) > P(বাজি) and also P(ফোন) > P(আছে). So the next word will be ফোন. After getting মোৰ ফোন, the probability of getting বাজি will be greater than that of আছে. So the next word will be বাজি. After মোৰ ফোন বাজি, the probability of getting আছে will be 1 (one), as there is no remaining word. i.e. P(আছে)=1. Therefore, আছে will be put next to get the full translated sentence [12]. There is a provision of getting equal probability also. In that case any one of the equi-probable words can be put. For a single sentence of a particular language, there may have more than one translated sentence. In that case, word order plays an important role in handling the situation.

- **Word order**

Assume that in the above case, obtained translated sentences are মোৰ ফোন বাজি আছে, মোৰ বাজি আছে ফোন, মোৰ আছে ফোন বাজি etc.. In this situation, there will be an ambiguity in producing the output. To handle this, the analysis of the text corpus is necessary as the appropriate order of the words to be placed in the sentences of a particular language can be learned from there. From the above three sentences, মোৰ ফোন বাজি আছে has the appropriate order of each words compared to মোৰ বাজি আছে ফোন and মোৰ আছে ফোন বাজি. Hence, it is expected to get মোৰ ফোন বাজি আছে as the ultimate output [19]. Other than this problem, there is a chance to encounter another problem i.e. getting more than one meaning of a single word. In this case, word choice resolves the issue.

- **Word choice**

In the previously considered sentence, the word *phone* can have many synonymous words in Assamese like ফোন, দূৰভাষ, দূৰালাপ, দূৰভাষণী, টেলিফোন etc.. Therefore, it is possible to get many translations based on these words. Since, each and every word in the text corpus has a specific probability; the choice of the word is done for the maximum probable word out of all the choices available on that corpus [19].

During the probability calculations in different steps of translation, *data sparsity* creates a major problem. Due to data sparsity, most possible word sequences may not be observed in training phase. As a result, the testing phase may show low efficiency. One solution is to make an assumption that the probability of a word depends only on the previous *n* words and this is called as n-gram model. The following section enlightens the n-gram model.

- **N-gram**

An *n*-gram can be termed as a contiguous sequence of *n* number of items from a particular sequence of text or speech [14]. Different applications may be having different types of items like phonemes, syllables, letters, words, base pairs etc.. An *n*-gram with size 1 is labeled as *unigram*, size 2 is labeled as *bigram* or *digram*, size 3 is labeled as *trigram* [14]. After these, the n-grams can be labeled as per the value of *n* like *four-gram*, *five-gram*, *six-gram* etc. [14], [9]. One example to understand this concept is given in Table- I.

○ ***N-gram models***

An *n*-gram model is based on the calculation of the probability and it is used to predict the next item for a sequence in the form of *Markov model* of order *(n - 1)*.

The n-gram models and the algorithms that use the n-grams have the following two main advantages:

- These are relatively simple and
- These are having the ability to scale up. It can be efficiently scaled up by simply increasing the value of *n* [12].

To make it clearer we can say that an n-gram model predicts Y_j by looking into $Y_{j-(n-1)}, \dots, Y_{j-1}$. Probability term for this is $P(Y_j | Y_{j-(n-1)}, \dots, Y_{j-1})$. Independent assumptions are made during the language modeling so that each word of a particular sentence is dependent only on the last $n-1$ words. These assumptions are important because:

- It significantly simplifies the problem associated with learning the language model from the data.
- The nature of the language is open and it is obvious to group the words together that are unknown to the language model.

It is to be noted that, in basic n-gram language model, the probability of a word from a sentence, with respect to some number of previous words from the same sentence (one word for bigram model, two words for trigram model, etc.) can be described as a *multinomial distribution* or *categorical distribution* [14].

Table- I: Demonstration of n-gram (up to 3-gram)

Type	Sample Sequence	1-gram sequence (unigram)	2-gram sequence (bigram or digram)	3-gram sequence (trigram)
Character	be_in_the_debt_of	b, e, _, i, n, _, t, h, e, _, d, e, b, t, _, o, f	be, e_, _i, in, n_, _t, th, he, e_, _d, de, eb, bt, t_, _o, of	be_, e_i, _in, in_, n_t, _th, the, he_, e_d, _de, deb, ebt, bt_, t_o, _of
Word	be in the debt of	be, in, the, debt, of	be in, in the, the debt, debt of	be in the, in the debt, the debt of

The probability $P(W_1, \dots, W_k)$, observing the sentence $W_1 \dots W_k$ is approximated in n-gram model as:

$$P(W_1, \dots, W_k) = \prod_{j=1}^{k} P(W_j | W_1, \dots, W_{j-1}) \approx \prod_{j=1}^{k} P(W_j | W_{1-(n-1)}, \dots, W_{j-1}) \quad (1)$$

where, W_k is the k^{th} word of the given sentence.

Using the n-gram frequency counts, the conditional probability can be calculated as follows:

$$P(W_j | W_{j-(n-1)}, ., W_{j-1}) = \frac{count(W_{j-(n-1)}, \dots, W_{j-1}, W_j)}{count(W_{j-(n-1)}, \dots, W_{j-1})} \quad (2)$$

This can be clarified with the help of the following example:

Let us consider a sentence *"I visited Gauhati University"*. The probability of this sentence using bigram $(n-2)$ can be approximated as-

P(I,visited,Gauhati,University)=P(I|<s>)P(visited|I)P(Gauhati|visited)P(University|Gauhati)P(</s>|University)

The approximation for the same sentence in trigram $(n = 3)$ is-

P(I,visited,Gauhati,University)=P(I|<s>,<s>)P(visited|<s>,I)P(Gauhati|I,visited)P(University|visited,Gauhati)P(</s>|Gauhati,University)

It can be noted that, the first $(n-1)$ n-grams contexts are filled with $< s >$ known as start-of-sentence marker. Without the end-of-sentence marker $(</s>)$, the probability of a smaller sequence *I visited Gauhati* would always be higher than that of the longer sequence *I visited Gauhati University* [14].

○ ***Count smoothing***

During the counting of n-grams, we may encounter that some n-grams do not appear in the corpora. Those n-grams contribute 0 (zero) probability to the corpora. This may yield a value 0 (zero) while retrieving and if it is needed to multiply the probabilities with the n-grams having non-zero probabilities. To get rid of this, 1 (one) is added to the numerator and the number of types is added to the denominator. So, the new count will look like the following:

$$P(W_j | W_{j-(n-1)}, ., W_{j-1}) = \frac{count(W_{j-(n-1)}, \dots, W_{j-1}, W_j) + 1}{count(W_{j-(n-1)}, \dots, W_{j-1}) + t} \quad (3)$$

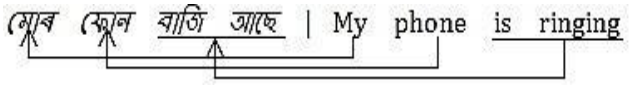

Fig. 4.Example of Sentence alignment

Where, t is the number of types present in the vocabulary. This makes sure that each n-gram has a minimum count of 1. Hence, a sequence that does not even occur in the corpora will also have a non-zero probability [19].

Phase-II: Translation model

Translation model aims to generate the sentence *d* of target language from the sentence *c* of source language by computing the conditional probability $P(d|c)$. It can be thought that the target sentence is being generated from source sentence word-by-word [12], [4]. The following example can be considered for better understanding:

If an English sentence *My phone is ringing* is translated to an Assamese sentence, one result is মোৰ ফোন বাজি আছে. One possible alignment for the sentence pair can be as in Fig. 4. A number of alignments are possible for every sentence, such as word-by-word, phrase etc. For better understanding, the word-by-word alignment can be considered. If m numbers of words are present in source sentence and n numbers of words are present in target sentence, then $m \times n$ different alignments are possible. All connections towards the target position are equi-probable. Therefore, the order of words in the target sentence and the source sentence do not affect the probability of target given source. i.e. $P(target|source)$ [20].

After word-to-word alignment, the tuples are extracted. The bilingual units are generally referred to as the tuple. The probabilities of the translation model are approximated by the n-grams of the tuples at sentence level as follows:

$$P(T,S) \approx \prod_{m}^{M} P((t,s)_k \,|(t,s)_{k-1},(t,s)_{k-2},\dots\dots,(t,s)_{k-n+1}) \quad (4)$$

Where, t represents target and s represents source. The $(t,s)_k$ refers to k^{th} tuple of bilingual sentence pair [15].

Phase-III: Decoding

The process of probability maximization for the translated text can be termed as the decoding. The words having maximum likelihood are chosen. A target sentence search is performed that maximizes $P(T|S)$. i.e. $P(S,T) = arg\ \max\ [P(S|T) \times P(T)]$ should be satisfied for getting the target sentence [12].

In the above formula, S and T denote source and target respectively. There is a possibility that a problem of infinite space search may arise in this case. To cope up with this problem, a method called *stacked search* is generally suggested. Here a partial alignment hypothesis list is maintained and the search is prompted to start with null hypothesis. That means we get the target sentence from an unknown source word sequence [12]. One example is as follows:

(অসমৰ ৰাজধানী দিছপুৰ | #), where, # is a place holder for an unknown source word sequence. As the search moves forward, one or more words are added to the hypothesis to extend the list of entries [12]. Example is as follows:

(অসমৰ ৰাজধানী দিছপুৰ | The Capital of Assam)

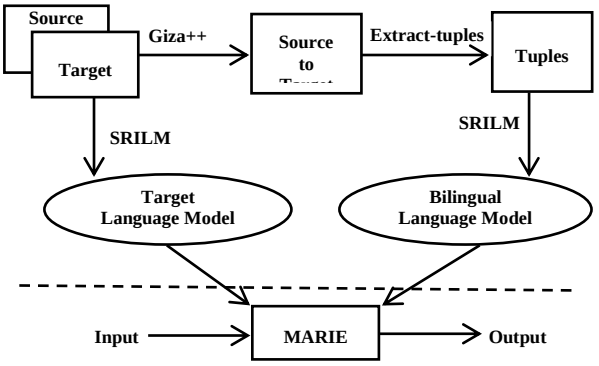

Fig. 5.Block diagram of the proposed system

The # can take any combination of the words in the list. But the search will terminate with a more promising complete alignment than any of the incomplete alignments [20].

B. *Architecture of the Proposed System*

The architecture of the proposed system (Fig. 5) has two parts. The upper part of the dotted line is for *training* and the lower part of the dotted line is for *testing* purpose. In the training phase the developed system is trained with the available data. In this case, the data is the English-Assamese parallel corpus. The more the data in the training phase, the better the efficiency of the system. Here, a corpus of about 15000 sentences has been used. From the architecture it can be said that the alignment of English words to Assamese words, the extraction of the tuples, creation of the target language model, creation of the bilingual language model and feeding these language models into the decoder falls under the training phase. The testing phase comprises the uses of 10 to 20 percentages of the training data to test the system, whether it is working properly or not. It is just an approximation that the system will work as desired. Basically, providing input to the decoder (MARIE) and getting output from the decoder falls under the testing phase.

From the block diagram (Fig. 5), it has been seen that parallel corpus of source and target language has been word-by-word aligned with the help of translation model toolkit called Giza++. From the aligned corpus the translation units are extracted. The translation units are called tuple in n-gram based statistical machine translation system. After extracting the tuples, the bilingual language model is created with the help of Stanford Research Institute Language Model (SRILM) toolkit. Also target language model is created with the help of SRILM from the target language corpus. The MARIE (n-gram based decoder) is then trained with the language models. After training, the decoder (MARIE) is ready to translate and accepts input from the user and yields the output accordingly. The details of the tools used in this work are described in the following section.

IV. EXPERIMENTAL SETUP

A. *Tools Used*

GIZA++ and SRILM have been used for translation model and language model respectively. For the decoding part MARIE has been used. The details regarding the tools have been given below:

GIZA++

In [16], it has been mentioned that GIZA was a part of the statistical machine translation toolkit called EGYPT. It was developed at the *Center for Language and Speech Processing* during the summer workshop in the year 1999 by the statistical machine translation team of Johns-Hopkins University [16]. GIZA was extended to GIZA++ by Fran Josef Och adding some new features [16]. The ideas about the extensions to GIZA which are included in GIZA++ can be found below:

- A new model named as *Model 4* [16].
- Another new model named as *Model 5* [16].
- Depending on word classes some Alignment models have been included [16].
- Implementation of the HMM alignment model: Baum-Welch training, Forward-Backward algorithm, empty word, dependency on word classes, transfer to fertility models etc. [16].
- New form for Model 3 as well as for Model 4 which allows the training of the parameter p_0 [16].
- For fertility, distortion or alignment parameters, many smoothing techniques have been included [16].
- Fertility models have been trained more efficiently than the previous ones [16].
- The pegging described in [4] has been accurately implemented. It is a series of heuristics to make the pegging efficient enough [16].

SRILM

It is a widely used language modeling toolkit. It calculates the n-grams of the corpus. The *n* may be of any value. Default value of n is 3. It requires huge monolingual corpus (In our case: Assamese) in well aligned manner. Also it can calculate the n-grams for bilingual corpus, such that the words are aligned and there is no gap between the aligned words.

The primary objective of SRILM is to support the *estimation* of the language model and *evaluation*. The creation of a model from the training data is termed as the estimation and the computation of the probability of a test corpus or corpora is termed as the evaluation.

The basis of SRILM is the n-gram statistics. Three main functionalities of SRILM are:

- To generate the count file of n-gram from the given corpus
- To train the language model with the help of the count file of n-gram
- To calculate the perplexity of the test data using the language model that has been trained.

For this system, SRILM has been used to get the target language model and also the bilingual language model. For target language model we have used Assamese text corpus.

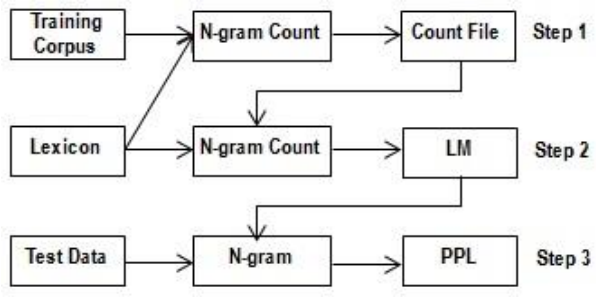

Fig. 6.Language model development steps

For bilingual language model we have used the file containing the translation units (tuple). These units are extracted from the alignment file of Giza++. After getting the tuples with the help of *extract-tuple* (a tool comes with MARIE), some modifications are needed to get the appropriate format for getting correct n-gram from the file. The SRILM tool calculates the n-gram in three steps as shown in the Fig. 6 [22].

MARIE

It is a statistical machine translation decoder based on the n-gram model. It implements the beam search strategy, which is based on the dynamic programming technique [15]. The decoding is guided by the source and it is performed monotonically [15]. While decoding, the partial translation hypotheses are generally arranged into distinct stacks as per the total number of source words they are covering [15]. The hypotheses covering the same source word competes with each other [15].

To tract the decoding at every step, MARIE uses *Threshold pruning* and *Histogram pruning* methods. In *Threshold pruning* method**,** each and every partial translation hypotheses having low score than a preset threshold value are eliminated [15], [7]. And in *Histogram pruning* method, the maximum number of partial translation hypotheses to be used is decided and the K-best ranked ones are considered [15]. *Hypothesis recombination* is also been allowed by MARIE. Here, exactly coinciding partial translation hypotheses in both the present tuple and the tuple trigram history are recombined [15].

B. *Steps Carried Out under the Proposed Work*

- **Creation and preprocessing of the corpus**

Parallel corpus of English and Assamese language is required for this translation system to work. Here, a parallel corpus of about 15000 sentences has been used. The domain of the corpus is mixed, comprising maximum part from tourism in India. The raw corpus may have some different format than the tool's format. That's why, before using the raw corpus, it is necessary to process it to get the desired format for the tools. If this is not done, the tools may not work properly and as a result it may not produce desired output. Some tools may need only the lowercase letters, some may need uppercase letters, some tools may require separate quotation mark etc.. Depending on the need of the tools, the data or the training data have to be processed first. Here, the following preprocessing steps are carried out:

- **Tokenization**

For correct alignment, the corpus should be in tokenized form. Tokenization means separation of the words, punctuations, extra spaces from the sentences of the corpus. This works by following some rule for the concerned language. The special rules are assigned in the *nonbreaking-prefix* file. For this purpose a perl script has been used. The perl script works only for some particular languages. So, for tokenizing Assamese corpus, some modification is needed. Since the structure of English and Assamese sentences is not same, hence direct use of the tokenizer results into break down of the composite words of Assamese sentence.

This is not desirable. Therefore, slight modification has been done into the code of the perl program to get the desired result for Assamese language. The commands for tokenization are:

./tokenizer.perl -l src<srccorpus>srccorpus.tok

./tokenizer.perl –l trg<trgcorpus>trgcorpus.tok

Here,

src→source language abbreviation (e.gen|as|hi|je|de etc.)

srccorpus → source corpus file along with the path

srccorpus.tok→ tokenized source file along with the path

trg → target language

abbreviation (e.gen|as|hi|je|deetc)

trgcorpus → target corpus file along with the path

trgcorpus.tok → tokenized target file along with the path

○ **Lower casing**

Lower casing of the words in the corpus is necessary for better result from Giza++. The uppercase letters are turned into lowercase letters using the inbuilt function of Ubuntu. The syntax for lower casing is:

tr '[:upper:]' '[:lower:]' <srcfile.tok>srcfile.tok.low

Here,

tr→ refers to translation, i.e translate from upper to lower case

[:xxxxxx:]→ this varies depending on the operation performed

srcfile.tok→ refers to tokenized file to be lowercased

srcfile.tok.low→ refers to output file after lowercasing.

Generally, this operation should be performed for both the source and the target file. But in Assamese language, there is no concept of lower and upper case. This language is not case sensitive. Yet this operation can be performed on the Assamese language corpus. This will not show any change in the corpus.

- **Creation of the translation model**

A major part of this system is the translation model. Basically, in this case, the calculation of the probability of the target sentence given the source sentence is performed. For this purpose Giza++ has been used. The details of creation of the translation model are discussed below:

○ **Translate the plain text to Giza format**

Giza++ works only on texts which are in Giza format. The plain text i.e. the tokenized and lowercased text is converted into Giza format with the help of the following:

./plain2snt.out srcfiletrgfile

Here,

srcfile→refers to the source language tokenized and lowercased file along with path

trgfile→refers to the target language tokenized and lowercased file along with path

This will create the files *srcfile.vcb*, *trgfile.vcb*, *srcfile_trgfile.snt*and*trgfile_srcfile.snt*. The descriptions of the files are as follows:

srcfile.vcb file contains the information from the source language corpus like, each word, frequency count and unique id.

trgfile.vcb file contains the information from the target language corpus like, each word, frequency count and unique id.

srcfile_trgfile.snt file contains each of the sentences from the parallel corpus (source and target) translated into the unique numbers for each of the words.

trgfile_srcfile.snt file contains each of the sentences from the parallel corpus (target and source) translated into the unique numbers for each of the words. It is just the opposite of *srcfile_trgfile.snt*and vice-versa is also correct.

○ **Create coocurrence file**

The coocurrence files are created from the above Giza format file. The syntax for creating the coocurrence file is as follows:

./snt2cooc.out srcfile.vcbtrgfile.vcbsrcfile_trgfile.snt >st.cooc

This will create the coocurrence file and write into the file *st.cooc*. The filename may be defined by the user i.e. it may be other than st.cooc.

○ **Make classes**

One package named as mkcls makes different classes of the text in the corpus based on similarity of the words. The number of class may be defined by the user. If the number of class is not defined, then the default number of class is created. The syntax of creating classes by mkcls is as follows:

./mkcls –C80 –n10 –psrcfile.tok.low –Vsrcfile.tok.low.vcb.classes

or

./mkcls –n10 –psrcfile.tok.low –Vsrcfile.tok.low.vcb.classes

The first command will create 80 numbers of classes using 10 numbers of rounds and the second command will create mkcls' default number of classes using 10 numbers of rounds. The number of rounds may vary. Higher the numbers of rounds better the result. This will create two files *srcfile.tok.low.vcb.classes* and *srcfile.tok.low.vcb.classes.cat* in the specified directory. The descriptions of the files created are as follows:

srcfile.tok.low.vcb.classes files hold the list of all words in alphabetical order along with the punctuations and also the frequency counts of all the words.

srcfile.tok.low.vcb.classes.cat files hold the list of frequencies and the collection of words for that corresponding frequency.

○ **Alignment of the corpus**

Once all the configuration files needed for the alignment are ready, the alignment process of the source and target corpus can be initiated. Since, .vcb files, .snt files, CoocurrenceFile and class files are obtained from earlier operations, Giza++ can align the corpus now. This is done using the following:

./GIZA++ -S srcfile.tok.low.vcb –T trgfile.tok.low.vcb –C srcfile_trgfile.snt –CoocurrenceFilesrc_trg.cooc

The above will align the source and target corpus with the help of source vcb, target vcb, source and target snt file and the coocurrence file. The files created by this are described below:

.a3.final file has a table with the below mentioned format:

$x\ y\ z\ t\ P(x|y,z,t)$

where,

x →Source sentence position

y →Target sentence position

z →Source sentence length

#Sentence pair (1) source length 19 target length 11 alignment score: 7.38552e-16
ভাৰতবৰ্ষৰ গোলপীয়া মহানগৰ নামে খ্যাত জয়পুৰ , ৰাজস্থান ৰাজ্যৰ ৰাজধানী।
NULL ({ }) ({ 1 2 3 4 5 }) Jaipur ({ 6 }) , ({ 7 }) popularly ({ }) known ({ }) as ({ }) the ({ }) pink ({ }) city ({ }) city ({ }) , ({ }) is ({ }) the ({ }) capital ({ 10 }) of ({ }) rajasthan ({ 8 }) state ({ 9 }) , ({ }) india ({ }) . ({ })
#Sentence pair (2) source length 22 target length 19 .....
.....

Fig. 7.Alignment Probability (.A3.final) file

t →Target sentence length

$P(x|y,z,t)$ →The

probability that a source word in the position x is moved to the position y in a sentence pair with length $\boldsymbol{z}$ and $\boldsymbol{t}$[18]

.A3.final contains-

The matches for the source sentence to the target sentences and give the match an alignment score. It is the desired alignment file. It creates three lines for a particular sentence. It shows the number of words present in the source sentence and also in the target sentence along with the alignment score (Fig. 7).

Source corpus words, their unique id and frequency counts are shown in *.trn.src.vcb*. It matches with *srcfile.vcb*.

Target corpus words, their unique id and frequency counts are shown in *.trn.trg.vcb*. It matches with *trgfile.vcb*.

*.tst.src.vcb*is empty.

*.tst.trg.vcb*is empty.

.perp file contains the list of perplexity for each iteration and model. The form of the file is as follows:

"trsz tssz itr mdl tr_pp ts_pp tr_vit_pp ts_vit_p"

where,

- *trsz*→size of training data
- *tssz*→ size of test data
- *itr*→ no. of iteration
- *mdl*→model
- *tr_pp*→perplexity of the training
- *ts_pp*→perplexity of the test
- *tr_vit_pp*→viterbi perplexity of the training
- *ts_vit_pp*→viterbi perplexity of the test

.d3.final is alike *.a3.final* but the position of x and y are switched.

.n3.final is having the probability of all the source tokens with 0 fertility, 1 fertility,……., N fertility.

The table created after all the iterations of Model 4 training is kept in *.t3.final*.

The probability for the insertion of null after a source word is kept in *.p_03.final*.

The translation table associated with Model 4 is kept in *.d4.final*.

The distortion table associated with IBM-4 is kept in *.D4.final*.

The parameter settings used in the training are stored in *.gizacfg*.

○ **Extraction of the translation units (tuples)**

For n-gram based translation system, translation units (often called as tuple) are extracted. A tool called as *extract-tuples* comes with MARIE. This is used to extract tuples from the aligned file. i.e from the *.A3.final* file. But this tool cannot provide the tuples in correct format as the SRILM tool needs to create the bilingual language model. This extracts the tuples as follows [6]:

srcword # trgword # 0

where,

srcword→ denotes the source word

trgword→ denotes the target word

The # between the words experiences two spaces before and after. Also an extra # 0 comes with it. This is not desirable. If we apply SRILM to count the n-gram on this type of file, then SRILM will break this into srcword, <space>, #, <space>, trgword, <space>, #, <space>, 0. And will calculate the n-gram for all of these. To get rid of this, some modification in the file is needed. The "<space>#<space>0" has been removed and also the "<space>#<space>" has been replaced by "#". Then n-gram has been calculated for the file (Fig. 8).

- **Language model creation**

After getting the translation model, creation of language model is necessary for training the system. Without the language model, working of statistical machine translation system is not possible. Creation of language model means calculation of the n-gram of each word in the corpus. For this system a tool called SRILM has been used to calculate the n-gram of the words in the corpus. The n-gram has to be calculated using the following:

./ngram-count –order n –text file1 –lm file2

or

./ngram-count –text file1 –lm file2

The first one calculates the n-gram of order n. The value of n is user defined. The second one does not have the order field. In this case SRILM will calculate up to 3-gram. 3-gram is the default value of SRILM. *file1* is the input file and the *file2* is the output file, i.e the language model file. The format of the output file is ARPA (Advanced Research Projects Agency) [22].

Two types of language model have been created in this case. These are: *Target language model* and *Bilingual language model*. These models are discussed below.

○ **Target language model**

#ভাৰতবৰ্ষৰ গোলপীয়া মহানগৰ নামে খ্যাত
jaipur#জয়পুৰ
the#NULL
capital of rajasthan state#ৰাজস্থান ৰাজ্যৰ ৰাজধানী
india#NULL
.#।
is#NULL
famous for its majestic forts , palaces and beautiful lakes#এই চহৰখন সুবৃহৎ কিল্লা , অট্টালিকা আৰু ঘূনীয়া হ্ৰদবোৰৰ বাবে বিখ্যাত
.....
.....

Fig. 8.Extracted tuples after modification

Target Language model is the language model created using the target language corpus. In this case it is Assamese.

The corpus is first tokenized and lowercased then target language model is created using SRILM. As it is mentioned previously that Assamese is not a case sensitive language, therefore, lowercasing this corpus can be skipped.

○ **Bilingual language model**

Bilingual Language model has been created using the file containing the tuples extracted from the aligned corpus. The parallel corpus of English and Assamese has been processed using Giza++ to get the desired aligned file. The aligned file is then processed through the extract-tuples and the tuples are extracted. Here, the extracted tuples were in the decoder's format. Therefore, one more step mentioned under the heading *Extraction of the translation units (tuples)* has been implemented to rectify this problem. After that the bilingual language model has been created.

- **Decoding**

After getting the language models, the decoder called MARIE has been trained. Once the decoder has been trained the system is ready to translate. The training of the decoder is very easy. Simply copy the language models to the directory of MARIE. Then take the test file of source sentences to be used as the input file for the system. The output is obtained by the following:

./marie-v1.1 –i inputfile –fBMbilingualLMfile

or

./marie-v1.1 –i inputfile –fBMbilingualLMfile –fTMtargetLMfile

The first one does not contain the target language model file. It yields the output based on the Bilingual Language model only. The words in the input file are matched by searching the Bilingual Language model file. The second one contains the Bilingual Language model file and the Target Language model file. The use of target language model file can help in producing better result. Decoding part is mainly to produce the result. The produced result is discussed in the *results and discussions* section.

V. RESULTS AND DISCUSSIONS

The system was trained with almost 15000 sentences from parallel corpus of English and Assamese. In the testing phase, 15% of the training data have been used. Table - II shows some of the translated sentences.

Some sentences are correctly translated and some of them are partially translated. The system has been tested using four orders in the SRILM. The order means the level up to which the n-gram is being calculated during the language model creation by SRILM. The output was produced by using only Bilingual Language Model (BLM) and also by Bilingual Language Model and Target Language Model (TLM) together. For all those produced output, BLEU score has been calculated and the variation is shown in the Fig. 9. It has been seen that, for orders 1, 2 and 3, the system performs same for only BLM and also for BLM & TLM together. But for the order 4, translation using only BLM is performing better than that of the translation using BLM and TLM together.

The system is able to translate the English text to Assamese text, but the translated Assamese text is not up to the mark of desired translation. The problems we have noticed in the translated text are as follows:

- Some words of the input sentence are not translated at all
- Some proper nouns are not translated

The first problem arises because of the alignment of some words to NULL. It is due to the high dissimilarity in the formation of the sentences and smaller corpus. Secondly, some of the proper nouns are translated as desired but some

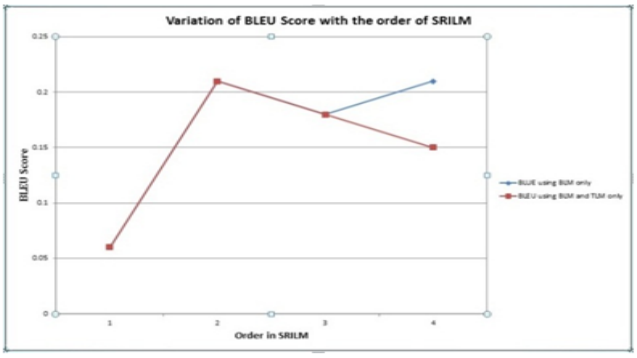

Fig. 9.Variation of BLEU score with the order of SRILM

are left out. One possible solution to the left out proper nouns is to translate the letters directly like A→অ, B→ব, C→চetc.

By increasing the training data set, it can be expected that the system will produce better result than the current result.

Table- II: Output from the user defined input

Input Sentence	Output Sentence
Mizoram is a land of great natural beauty.	মিজোৰাম প্ৰাকৃতিক সৌন্দৰ্যৰ ৰাজ্য।
Khimsar is a fine tourist destination in Rajasthan.	Khimsar ৰাজস্থানৰ এখন সুন্দৰ পৰ্যটনস্থলী।
Mumbai is a cluster of seven islands.	মুম্বাই সাতটা দ্বীপৰ সমষ্টি।

VI. CONCLUSION

The demand for the machine translation is increasing day-by-day. The machine translation for English-Assamese language pair is lagging behind the other language pairs. Assamese speaking peoples of northeastern India are not able to access many facilities due to language incomprehensibility. Keeping this in mind, a noble approach to translate English text to Assamese text using n-gram model has been initiated. Though this system has also been developed using phrase based decoder, yet the same system has been developed using different decoder hoping to get a better system in terms of the speed and less memory consumption. The performance and quality of translation vary with the change of used method and the dataset. In recent research, the accuracy of translation has shown significant improvement with the use of neural network and deep learning techniques.

ACKNOWLEDGMENT

The English-Assamese parallel corpus has been provided by the department of Information Technology, Gauhati University, Gopinath Bordoloi Nagar, Jalukbari, Guwahati, Assam, India.

REFERENCES

1. M. V. M. Barkade, P. R. Devale, "English to Sanskrit machine translator (lexical parser)", International Journal on Computer Science and Engineering, vol. 02, no. 06, pp. 2084-2091, 2010.
2. A. K. Barman, J. Sarmah, S. K. Sarma, "Assamese WordNet based Quality Enhancement of Bilingual Machine Translation System", in Proceedings of the Seventh Global Wordnet Conference, University of Tartu press, Jan. 2014, pp. 256-261.
3. K. K. Baruah, P. Das, A. Hannan, and S. K. Sarma, "Assamese-English bilingual machine translation", International Journal on Natural Language Computing, Available: https://doi.org/10.5121/ijnlc.2014.3307.
4. P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation", Association for Computational Linguistics, vol. 19, no. 2, pp. 263-311, 1993.
5. J. M. C. Crego, J. B. M. Acebal, "Architecture and modelling for n-gram-based statistical machine translation", Ph.D dissertation, TALP Research Center, Speech Processing Group, Department of Signal Theory and Communications, Universitat Polit`ecnica de Catalunya, Barcelona, 2008.
6. J. M. Crego, M. R. Costa-jussa, J. B. Marino, and J. A. R. Fonollosa, "Ngram-based versus phrase-based statistical machine translation", TALP Research Center, Universitat Polit`ecnica de Catalunya, Barcelona, 2014.
7. J. M. Crego, J. B. Marino, and A. de. Gispert, "An ngram-based statistical machine translation decoder". TALP Research Center, Universitat Polit`ecnica de Catalunya, Barcelona, INTERSPEECH, 2005.
8. J. M. Crego, F. Yvon, "Factored bilingual n-gram language models for statistical machine translation", Mach Translat, Available: https://doi.org/10.1007/s10590-010-9082-5.
9. P. Das, K. K. Baruah, "Assamese to English statistical machine translation integrated with a transliteration module", International Journal of Computer Applications, vol. 100, no. 5, pp. 0975-8887, 2014.
10. P. Das, K. K. Baruah, A. Hannan, S. K. Sarma, "Rule based machine translation for Assamese-English using apertium", International Journal of Emerging Technologies in Computational and Applied Sciences, vol. 8, no. 5, pp. 401-406, 2014.
11. G. V. Garje, G. K. Kharate, "Survey of Machine Translation Systems in India", International Journal on Natural language Computing, Available: https://doi.org/10.5121/ijnlc.2013.2504.
12. A. Hannan, S. K. Sarma, Z. Hussain, "Marie: A statistical approach to build a machine translation system for English Assamese language pair", International Journal of Computer Sciences and Engineering, Available: https://doi.org/10.26438/ijcse/v7i3.774779.
13. J. Hutchins, "Multiple uses of machine translation and computerised translation tools", International Symposium on Data and Sense Mining, Machine Translation and Controlled Languages, 2009.
14. D. Jurafsky, J. H. Martin, "N-gram language models". In (Third Edition draft), Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, pp. 37-62, 2018.
15. J. B. Marino, R. E. Banchs, J. M. Crego, A. de. Gispert, P. Lambert, J. A. R. Fonollosa, M. R. Costa-jussa, "N-gram-based machine translation", Association for Computational Linguistics, vol. 32, no. 4, pp. 527-549, 2006.
16. F. J. Och, H. Ney, "A systematic comparison of various statistical alignment models", Computational Linguistics, vol. 29, no. 1, pp. 19-51, 2003.
17. M. D. Okpor, "Machine translation approaches: Issues and challenges", International Journal of Computer Science, vol. 11, no. 5, pp. 159-165, 2014.
18. M. A. Sati, "Word alignment using Giza++ and Cygwin on windows", International Journal of Engineering Research & Technology, vol. 2, no. 5, pp. 1762-1765, 2013.
19. M. T. Singh, R. Borgohain, S. Gohain, "An English-Assamese machine translation system", International Journal of Computer Applications, vol. 93, no. 4, pp. 1-6, 2014.
20. N. Sharma, P. Bhatia, V. Singh, "English to Hindi statistical machine translation system", ME thesis, Thapar University, Patiala, 2011.
21. J. Slocum, "A survey of machine translation: Its history, current status, and future prospects", Computational Linguistics, vol. 11, no. 1, pp. 1-17, 1985.
22. A. Stolcke, "SRILM-An extensible language modeling toolkit", Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, U.S.A., 2004.
23. P. Unnikrishnan, P. J. Antony, Dr. K. P. Soman,"A novel approach for English to South Dravidian language statistical machine translation system", International Journal on Computer Science and Engineering, vol. 02, no. 08, pp. 2749-2759, 2010.

AUTHORS PROFILE

Zakir Hussain is a research scholar in the department of Computer Science and Engineering at National Institute of Technology (NIT) Silchar. He has completed his M.Tech. in Information Technology from Gauhati University in the year 2015. He has completed his BE (Honours) in Computer Science and Engineering in the year 2012 and bagged 4th position in the university. He was the state topper in the subject of General Science in board examination conducted by Assam state board in the year 2005.He was a Guest/Part time faculty in the level of Assistant Professor at Barak Valley Engineering College; a government of Assam institution in the district of Karimganj, Assam. He has 4+ years of teaching and administrative experiences in government run educational institutions and offices. His research interest is in the field of Natural Language Processing, Machine Learning, Data Analytics, and Data Mining.

Dr. Malaya Dutta Borah is working as an Assistant Professor in the Department of Computer Science & Engineering at National Institute of Technology (NIT) Silchar, Assam. She has completed her PhD in Computer Science and Engineering from Delhi Technological University (Formerly Delhi College of Engineering), Delhi. She has completed her M.E. (with distinction) in Computer Technology and Applications from Delhi College of Engineering, Delhi and B.Tech. in Computer Science and Engineering from North Eastern Regional Institute of Science and Technology, Arunachal Pradesh. Before joining NIT, she worked at Assam Engineering College, Delhi Technological University, Inderprastha Engineering College. She has authored/co-authored around 30 research papers in national/international journals/Conferences. She is actively involved in research works in the field of Data Mining, Cloud Computing, ICT and e-governance. She is the associate member of CSI (India) and IEEE. For more information, kindly visit: http://cs.nits.ac.in/malaya/

Abdul Hannan has completed his MSc. in Computer Science from Gauhati University in the year 2009 and M.Tech. in Information Technology from Gauhati University in the year 2012. He is currently working as a faculty in Gauhati University. He is pursuing his PhD in Machine Translation for Assamese Language. He has 7+ years of teaching experience in various educational institutions. His area of Interest is NLP, Computer Networks, and Machine Learning.