International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

Marie: A Statistical Approach to Build a Machine Translation System for English Assamese Language Pair

Abdul Hannan[1*], Shikhar Kr. Sarma[2], Zakir Hussain[3]

[1,2]Department of Information Technology, Gauhati University
[3]Department of Computer Science and Engineering, NIT Silchar

Corresponding author: hannan.guniv@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i3.774779 | Available online at: www.ijcseonline.org

Accepted: 23/Mar/2019, Published: 31/Mar/2019

***Abstract*—**The demand of Machine Translation (MT) is increasing due to the increased rate of exchange of information around the globe. Considering Internet as the main channel of information sharing, the source of information is not confined to a specific geographical location and a specific language. MT is the way of translating from one language to another with the help of computer system. The text of source language fed to the system and the system translates it to the target language. Many approaches and tools for those approaches have been developed to achieve better performance in translation. In this paper an n-gram based statistical approach is discussed.

***Keywords*—**Machine Translation, Marie, SMT,n-gram

I. INTRODUCTION

Machine Translation, a subfield of computational linguistics which is again a subfield of Artificial Intelligence is one of the most vital researches in the field. MT denotes translating text from one natural language to another natural language by following certain rules or statistical measures. In the Internet English is occupying near about 53.1% content. For a non-English speaking person, it is difficult to extract the information from more than half of the total information available in the Internet. Also the reverse is another worse situation as contents written in a native language is hardly understood by a non-native person. In these situations, machine translation can be the ultimate solution. The idea of machine translation system came during early 17th century. A universal language was proposed by Rene Descartes in 1629, where thoughts of different languages can be expressed using a single set of symbols. In 1949, in Memorandum on Translation of Warren Weaver the field machine translation appeared. The very first research on this field started at MIT by Yehosha Bar-Hillel in 1951. Later on a team of Georgetown university's MT researchers started working on it 1954. Eventually, in 1955 research started in Japan and Russia. In London the first conference on Machine Translation was held in 1956. SYSTRAN first started providing Machine Translation over the web in 1996. Other translation system also became available such as AltaVista's Babelfish and Google Language Tool, both of them used SYSTRAN Technology [3].

II. DIFFERENTAPPROACHES

Different approaches have been developed to achieve better result in machine translation. Typically, three translation approaches are distinguished: *direct approach*, *transfer approach* and *Interlingua approach* [8].

A. In this approach linguistic analysis of the source sentence is not done in order to produce a target sentence. Translation is done word-by-word basis. In early Machine Translation (MT), this approach was used. Now-a-days this has been abandoned even in the framework of corpus-based approaches.

B. Transfer approach is divided into three steps: analysis, transfer and generation. The source sentence is analyzed by producing an abstract representation. The transfer step transfers the abstract representation of first step into a corresponding representation in the target language. The generation step produces the target sentence from the intermediate representation.

C. The Interlingua approach produces a thorough syntactic and semantic analysis of the source sentence, turning the translation task into generating a target sentence according to the obtained Interlingua representation. This involves the deepest analysis of the source sentence. The advantage of Interlingua approach is that once the meaning of the source sentence is captured, it can be expressed in any number of target languages. All the above three can be shown by the figure1.

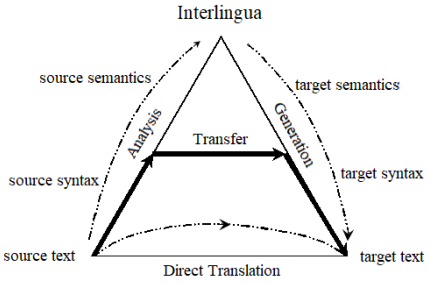

Figure1. Machine Translation Pyramid

Machine Translation systems can also be classified on the basis of the core technology they use. For example;

1. Rule Based: In case of rule-based approach, human experts specify some rules, to describe the translation process [5]. Different human experts may specify different rules for translation process. So, for different person, the system will be of different configuration and of different efficiency. This approach directly conveys the work of human experts.
2. Corpus-based: Here, the knowledge is automatically extracted by analyzing translation examples from a parallel corpus. Once the required technique has been analyzed for a given pair of languages, the translation system can be developed very quickly. A corpus-based approach generally follows direct or transfer approach [7]. The corpus-based approach can further be divided into two categories: example-based and statistical.
   - Example-Based MT (EBMT) uses the examples from the parallel corpora. Translation is provided by choosing and combining the examples.
   - Statistical MT (SMT) uses the examples from the parallel corpus to train the statistical translation system. This approach initially worked only on word-by-word basis. But now-a-days, it attempts to introduce certain degree of linguistic analysis.

In this paper we are showing the Statistical translation system for English-Assamese language pair. There are three basic components required for SMT, a Language Model (LM), Translation Model(TM) and Decoder.

Language model involves the calculation of the probability of each word in the target language corpus. Translation modelling involves the calculation of the probability of words in the target language corpus given the probability of the source language corpus. The decoding phase involves the maximization of the probability to get the correct translation.

In our work Marie decoder has been used to develop *n*-gram based Statistical Machine Translation system. To develop language model (LM) and word alignment SRILM and GIZA++ tools are used. For evaluation of bilingual translation Bilingual Evaluation Understudy (BLEU) score is used.

A key challenge faced during our work is OOV (out of vocabulary) words. The OOVs are ignored by the statistical machine translation system, which are later taken care of by introducing transliteration system.

Transliteration is the procedure which consists in representing the characters of one script by the characters of another and the operation remains reversible.

Assamese (অসমীয়া) language is one of the major languages spoken in northeast India, by roughly 19 million speakers mainly in Assam. Writing direction of Assamese language is left to right like English.

A. Language Model

Language modelling is the process of calculating the probability of occurrence of a word after another word in a particular language.

Suppose an English sentence *"The capital of Assam is Dispur"*. The translated Assamese sentence can be *"*অসমৰ ৰাজধানী দিছপুৰ*"* or "দিছপুৰ অসমৰ ৰাজধানী". For the first translated sentence, after placing "অসমৰ", the next word to be placed is; either "ৰাজধানী" or "দিছপুৰ". During the probability calculation if *P(*ৰাজধানী*) >P(*দিছপুৰ*)* then the word "ৰাজধানী" will be placed else "দিছপুৰ" will be placed. We assumed *P(*ৰাজধানী*) >P(*দিছপুৰ*)*, so the word *"*ৰাজধানী" is placed after "অসমৰ". After getting "অসমৰ ৰাজধানী", the probability of getting "দিছপুৰ" will be 1 (one), because there is no remaining word. i.e. *P(*দিছপুৰ*)*=1. So "দিছপুৰ" will be put next and will get the full translated sentence as "অসমৰ ৰাজধানী দিছপুৰ"

The second sentence will also follow the same rule. There is a provision of getting equal probability. In that case any one of the equiprobable words can be put. In the above case we will first get *P(*অসমৰ*)=P(*দিছপুৰ*)*, given the start. Hence any one of the two words can be put first. After putting the first word the above rule will be followed. The ambiguity can also be broken by assigning some rule.

- *Word Order*

Since in the above case we have got two translated sentence, hence there will be conflict which one will be given as output. In that case by analyzing the text corpus, the probability of both the sentences is calculated. If we get *P(*অসমৰ ৰাজধানী দিছপুৰ*) > P(*দিছপুৰ অসমৰ ৰাজধানী*)* then "অসমৰ ৰাজধানী দিছপুৰ" will be given as output. Otherwise the "দিছপুৰ অসমৰৰাজধানী" will come out as output. [2]

- *Word Choice*

Again there may have several meaning of a word. In the above case, "Capital" has many translations in Assamese language. Some of these are "ৰাজধানী", "মূল", "আচল" etc. So we may get following sentences as the translation.

অসমৰ ৰাজধানী দিছপুৰ or দিছপুৰ অসমৰ ৰাজধানী

অসমৰ মূল দিছপুৰ or দিছপুৰ অসমৰ মূল

অসমৰ আচল দিছপুৰ or দিছপুৰ অসমৰ আচল

In this case the probability of all the sentences is calculated by analyzing the text corpus. Since each word in the text corpus has a specific probability, the probability of the sentences will vary. The sentence with highest probability is given as the output. [2]

- *N-gram models*

An *n*-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a *(n - 1)* order Markov model.

The two core advantages of *n*-gram models (and algorithms that use them) are

- Relative simplicity and
- The ability to scale up by simply increasing *n*, a model can be used to store more contexts with a well understood space, time trade off, enabling small experiments to scale up very efficiently.

More concisely, an *n*-gram model predicts x_i based on $x_{i-(n-1),\ldots\ldots,}x_{i-1}$.In probability terms, this is $P(x_i|x_{i-(n-1),\ldots\ldots,}x_{i-1})$. When used for language modelling, independence assumptions are made so that each word depends only on the last *n-1* words. This Markov model is used as an approximation of the true underlying language. This assumption is important because it massively simplifies the problem of learning the language model from data. In addition, because of the open nature of language, it is common to group unknown words to the language model together.

In an *n*-gram model, the probability $P(w_1,\ldots\ldots,w_m)$ of observing the sentence $w_1\ldots\ldots w_m$ is approximated as

$$\boldsymbol{P(w_1,\ldots w_m)} = \prod_{i=1}^{m} \boldsymbol{P(w_i|w_1,\ldots w_{i-1})} \approx \prod_{i=1}^{m} \boldsymbol{P(w_i|w_{1-(n-1),\ldots} w_{i-1})}$$

Here, it is assumed that the probability of observing thei^{th}word $w_i$in the context history of thepreceding *i-1* words can be approximated by the probability of observing it in the shortened context history of the preceding *n-1* words (*n^{th} order Markov property).*

The conditional probability can be calculated from *n*-gram frequency counts:

$$\boldsymbol{P(w_i|w_{1-(n-1),\ldots} w_{i-1})} = \frac{\boldsymbol{count(w_{i-(n-1),\ldots.} w_{i-1}, w_i)}}{\boldsymbol{count(w_{i-(n-1),\ldots} w_{i-1})}}$$

The words bigram and trigram language model denote *n*-gram language models with *n*=2 and *n*=3, respectively.

Note that the context of the first *n-1 n*-grams is filled with start-of-sentence and end-of-sentence markers, typically denoted by<s> and </s> respectively.

Additionally, without an end-of-sentence marker, the probability of an ungrammatical sequence would always be higher than that of the longer sentence. [3]

- *Count Smoothing*

During the counting of *n*-grams, there is a possibility that some *n*-grams may not appear in the corpora. This will provide the probability of 0 (zero) to the corpora. This may produce a value 0 (zero) after multiplying the probabilities of the *n*-grams. To take care of this, smoothing is done. Here 1 (one) is added to such *n*-grams. So, there will not be any *n*-gram with count 0 (zero). The new count will be as follows:

$$\boldsymbol{P(w_i|w_{1-(n-1),\ldots} w_{i-1})} = \frac{\boldsymbol{count(w_{i-(n-1),\ldots.} w_{i-1}, w_i)} + \mathbf{1}}{\boldsymbol{count(w_{i-(n-1),\ldots} w_{i-1})} + \boldsymbol{t}}$$

Where, *t* is the number of types in the vocabulary. This ensures that each *n*-gram has at least a count of 1. So, sequence that does not occur will have a non-zero probability [2].

B. Translation Model

The aim of translation model is to generate the target language sentence *y* from the source language sentence *x* by computing a conditional probability P(*y*|*x*). The target sentence of the source sentence is thought of as being generated from source word-by- word.

Figure2. One possible alignment

A number of alignments are possible for every sentence. Such as word-by-word, phrase etc. For simplicity word-by-word alignment is considered. If the length of source is *m* and target is *n*, then there is *m*×*n* different alignments are possible and all connections for each target position are equiprobable. Therefore, order of words in target and source does not affect probability calculations i.e. P(*target*|*source*).[4]

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

$$\boldsymbol{P(T,S)} \approx \prod_{k=1}^{m} \boldsymbol{P(t,s)_k|(t,s)_{k-1},(t,s)_{k-2},\ldots,(t,s)_{k-n+1})}$$

Where, *t* corresponds target and *s* corresponds source, and $(t,s)_k$ refers to k^{th}tuple of a bilingual sentence pair. [9]

C. Decoding

Decoding is the process of maximizing the probability of translated text. The words are chosen which have maximum likelihood. Search for a sentence *T* is performed that maximizes *P(S|T)*. i.e

$$P(S, T) = argmax\ [P(T)\ P(S|T)]$$

Where, *S* and *T* denote source and target respectively. Here problem of infinite space search arises. To get rid of this problem, stacked search is suggested, where a list of partial alignment hypothesis is maintained. Here, search starts with null hypothesis. That means the target sentence is obtained from a sequence of source words that we do not know. One example is as follows:

(অসমৰ ৰাজধানী দিছপুৰ| *)

Where * is a place holder for an unknown sequence of source words. With the proceeding of search, the entries in the list are extended by adding one or more words to the hypothesis. Example:

(অসমৰ ৰাজধানী দিছপুৰ|The Capital of Assam)

The search terminates with a complete alignment in the list that is more promising than any of the incomplete alignments. [4]

III. TOOLSUSED

To implement the system, many tools have been used for different purposes. For translation model and language model we have used GIZA++ and SRILM respectively. For the decoding part we have used MARIE.

A. GIZA++

GIZA++ is a word alignment toolkit. Using GIZA++ the words of two languages can be aligned.

B. SRILM

It stands for Stanford Research Institute Language Model. It is a widely used language modelling toolkit. It calculates the *n*-grams of the corpus. '*n*' may be of any value. Default value of *n* is 3. It requires huge monolingual corpus 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 [1].

The main objective of SRILM is to support language model estimation and evaluation.

- Estimation: create a model from training data
- Evaluation: compute the probability of a test corpus for which conventionally expressed as the test set perplexity.

SRILM is based on *n*-gram statistics. Three main functionalities are:

- Generate the n-gram count file from the corpus
- Train the language model from the n-gram count file
- Calculate the test data perplexity using the trained language model

For our system we have used this tool to get the target language model and also the bilingual language model. For target language model we have used Assamese text corpus. For bilingual language model we have used the file containing the translation units (tuple). These units are extracted from the alignment file of GIZA++ [6]. After getting the tuples with the help of extract-tuple, some modification is needed to get the appropriate format for getting correct *n*-gram from the file.

C. MARIE

It is an *n*-gram based statistical machine translation decoder. It is based on beam search. MARIE implements a beam-search strategy based on dynamic programming. The decoding is performed monotonically and is guided by the source. During decoding, partial-translation hypotheses are arranged into different stacks according to the total number of source words they cover. In this way, a given hypothesis only competes with those hypotheses that provide the same source word coverage. At every translation step, stacks are pruned to keep decoding tractable. MARIE allows for two different pruning methods:

- *Threshold pruning:* For which all partial-translation hypotheses scoring below a predetermined threshold value are eliminated.
- *Histogram pruning:* For which the maximum number of partial-translation hypotheses to be considered is limited to the *K*-best ranked ones.

Additionally, MARIE allows for hypothesis recombination, which provides a more efficient search. In the implemented algorithm, partial-translation hypotheses are recombined if they coincide exactly in both the present tuple and the tuple trigram history.

IV. DEVELOPMENT OF THESYSTEM

The English-Assamese translation system has been developed using the tools discussed. The step by step procedure of developing the system has been given below:

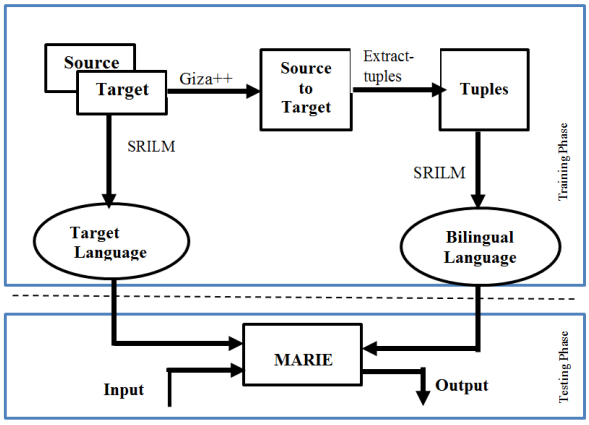

Figure3. System Architecture

Parallel corpus of Source and Target language has been word by word aligned with the help of Giza++. From the aligned corpus the translation units (tuple) are extracted.

After extracting the tuples, the Bilingual Translation Model is formed with the help of SRILM. Also Target Model is formed with the help of SRILM from the target language corpus.

The *n*-gram based decoder; MARIE is then trained with the language models. After training the decoder is ready to translate. MARIE then takes input from the user and gives the output accordingly.

The architecture has been divided into two parts- *training* and *testing*.

A. Training phase

In the training phase the system is trained with English-Assamese parallel corpus. Here we have used a corpus of about 15,000 sentences from tourism domain. So, from the architecture we can say that the alignment of English words to Assamese words, the extraction of the tuples, creation of the language model and creation of the bilingual language

model falls under the training phase. The following steps are followed in development and training phase

- *Preprocessing of the corpus*

Since the corpus we have used is in raw format, we need to process it to fit in our system. For our work the pre-processing steps are as follows:

– *Tokenization*

For correct alignment, the corpus should be of tokenized. Tokenization means separation of the words, punctuations, and extra spaces from the sentences of the corpus. Since the structure of English and Assamese sentences is not same, use of the common tokenizer results into break down of the composite words of Assamese sentence. This is not desirable. So, slight modification has been done for Assamese language.

– *Lower Casing*

Lower casing of the words in the corpus is necessary for better result of Giza++. The uppercase letters are turned into lowercase letters in this process. Generally, this operation should be performed for both the source and target file. But in Assamese language there is no lower and upper case. So we have done this phase only for English Sentences.

- *Translation Model creation*

Translation model is a major part of the system. Basically, in this case the calculation of the probability of the target sentence given the source language is performed. For this we have used Giza++.

i. *Translate the plain text to Giza format:* We convert the corpus to the required Giza format.
ii. *Create Co-occurrence File:* The Co-occurrence files are created from the output file of previous step with extension **.cooc*.
iii. *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 classes may be defined by the user. If the number of classes is not defined, then the default number of class is created.
iv. *Alignment of the corpus:* Alignment for both source and the target corpus is done. After alignment a **.final* file is created containing a table with the following format:

$$I\ J\ L\ M\ P\ \left(\frac{I}{J}, L, M\right)$$

Where,

I=position of the source sentence
J=position of the target sentence
L=length of the source sentence
M=length of the target sentence

$P\ \left(\frac{I}{J}, L, M\right)$=the probability that a source word in position *I* is moved to position *J* in a pair of sentences of length *L* and *M*.

v. *Extraction of the translation units (tuples):* For *n*-gram based translation system, translation units are extracted. A tool of MARIE itself called *extract-tuples* used to perform the extraction

- *Language model creation*

After getting the translation model, creation of language model is necessary for training of the system. Without the language model, statistical machine translation system cannot work. Creation of language model means calculation of the *n*-gram of each word in the corpus. The corpus may be unilingual or bilingual.

- *Get n-gram using SRILM*

For our system a tool called SRILM has been used to calculate the *n*-gram of the words in the corpus. Two types of language model have been created in our work. These are:

– *Target Language model:* Target Language model is the language model created using the target corpus. The corpus is first tokenized and then the n-grams have been calculated using SRILM.

– *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 then processed through the *extract-tuples* module and the tuples have been extracted. After the arrangement of the tuples, the file has been used to calculate the n-grams of the tuples.

- *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. While decoding it is better to use both bilingual and target language model, because, it can produce better word if possible.

Suppose, an input sentence *"Dispur is the capital of Assam"* has been translated into "দিছপুৰ অসমৰ মূল" as in the Bilingual Translation model. But we know that for this case the word "ৰাজধানী" is more appropriate than the word "মূল". The use of Target Language model along with the Bilingual Language model, the probability of getting "ৰাজধানী" in place of "মূল" is higher.

B. Testing phase

The testing phase comprises the uses of 15% 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. We have used 2075 sentences from tourism and mix domain for testing the system. The result produced by the developed system is discussed in result and discussion section.

V. RESULT AND DISCUSSION

As we have stated earlier, we have used parallel sentences from mainly tourism and other domain.

Some of the translated sentences are stated bellow-

i. *Gauhati University was established in 1948* → *গুৱাহাটী বিশ্ববিদ্যালয় ১৯৪৮ত স্থাপিত হৈছিল*

ii. *গুৱাহাটী উত্তৰপূৰ্ব ভাৰতৰ কেন্দ্ৰ* → *Guwahati is the center of northeast India.*

iii. *গুৱাহাটী বিশ্ববিদ্যালয় ১৯৪৮ চনত স্থাপিত হৈছিল* → *Guwahati University was established in the year 1948*

In the first line Gauhati is translated into গুৱাহাটী which is a correct translation for the name of the university. It is also seen in the second sentence where গুৱাহাটী is translated into Guwahati. But in the third sentence গুৱাহাটী বিশ্ববিদ্যালয় is translated into Guwahati University which is not correct. It is happening because in the training corpus Guwahati is appearing more number of times than that of Gauhati for Assamese word গুৱাহাটী. Based on the probability calculation Guwahati is more probable than Gauhati.

The result is seen not too much satisfactory but still acceptable. The system is also tested for different length of sentences. It is also seen that the quality of translation depends on the length of the source sentence. The overall BLEU score of the system is 0.21 which is not so poor but still way less than the state-of-the-art translation system. The BLEU score is also calculated in different dimensions. The BLEU scores for different length of sentences is plotted in a graph and shown in the figure bellow.

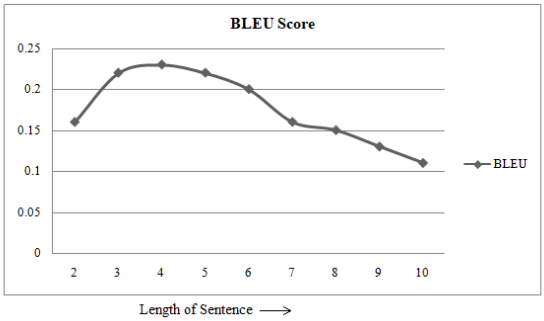

Figure4. Blue Score of different length of sentence

From the above result it is seen that as the length of the sentence increasing the quality of the translation is decreasing. Even though it is far away from the BLEU score of the state of the art translation system, this system is performing well with sentences of three to five words. It can be the cause of wrong word alignment. A better result can be expected with a bigger size in training data set. The system is able to translate the English text to Assamese text, but the translated Assamese text is not up to the mark. The problems we have noticed in the translated text are as follows:

- Some words of the input sentence are produced in the target language as it was in the source.
- The proper nouns are mostly ignored during translation.

The first problem is due to the alignment of the words to NULL. That means the target language word for the input word has been aligned to NULL. One solution to the problem is to assign a nearest word (preceding or succeeding) to the NULL aligned word. If we assign a word to the input sentence word, then during the extraction of tuple we will not get the NULL gained tuples.

By increasing the training dataset, it can be assumed that the system will work to give better result than current result for English Assamese language pair.

REFERENCES

[1] Andreas Stolcke,*"Srilm —An Extensible Language Modeling Toolkit"*, In the proceedings of International Conference on Spoken Language Processing, Vol. **2**, pp **901-904**, Denver, **2002**

[2] M. D. Okpor, *"Machine Translation Approaches: Issues and Challenges"*, IJCSI International Journal of Computer Science Issues, Vol. **11**, Issue **5**, No 2, September **2014**

[3] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, *"BLEU: a Method for Automatic Evaluation of Machine Translation"*, In the Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. **311-318**, July **2002**

[4] P. Brown, V. Pietra, S. Pietra, R. Mercer, *"The Mathematics of Statistical Machine Translation: Parameter Estimation"*, Journal of Computational Linguistics, Vol. **19**, No. **3**, **1993**

[5] Muhammad Naeem Ul Hassan, *"Urdu Language Translation using LESSA"*, International Journal of Scientific Research in Computer Science and Engineering Vol.**6**, Issue.**5**,**2018**

[6] Philip Koehn, "Moses, Statistical Machine Translation System, User Manual and Code Guide", University of Edinburgh, pp. **234-255**, **2019**

[7] Sandipan Dandapat, Sara Morrissey, Andy Way, Joseph van Genabith, *"Combining EBMT, SMT, TM and IR Technologies for Quality and Scale"*, In the proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, France, pp. **48–58**, **2012**

[8] Mohamed Amine Cheragui, *"Theoretical Overview of Machine translation"*, in the Proceedings of International conference on Web and Information Technologies, Algeria, pp. **160-169**, **2012**

Authors Profile

Abdul Hannan has pursued M.Sc. in Computer Science in year 2009, M.Tech. in Information Technology year 2012 from Gauhati University and pursuing Ph.D. from Gauhati University. He is working as a faculty in Gauhati University from 2012. His main area of research is Machine Translation, Computer Networks and Data Mining.

Dr. Shikhar Kumar Sarma is currently working as a Professor in the department of Information Technology, Gauhati University. He is a pioneer in the field of Computational Linguistics for Assamase languages. His domain of research is Natural Language Processing, Language Technology and Artificial Intelligence.

Zakir Hussain has pursued M.Tech. in Information Technology from Gauhati University in the year 2015. He is currently pursuing his Ph.D. from Dept. of Comp& Engineering, NIT Silchar.

