

# An English-Assamese Machine Translation System

Moirangthem Tiken Singh

Institute of Engineering and Technology,  
Dibrugarh University,  
Assam, India.

Rajdeep Borgohain

Institute of Engineering and Technology,  
Dibrugarh University,  
Assam, India.

Sourav Gohain

Institute of Engineering and Technology,  
Dibrugarh University,  
Assam, India.

## ABSTRACT

Statistical machine translation is the process of translating languages using machine learning methods. By learning from a large parallel text corpora, Statistical Machine Translation automatically learns how to translate one language into another. This paper will present a method to translate English language to Assamese language using Statistical Machine Translation.

## General Terms:

Machine Translation from one language to other

## Keywords:

Statistical Machine Translation, English Language, Assamese Language

## 1. INTRODUCTION

Machine translation is the automatic translation from one natural language into another using computer. Machine translation draws ideas and techniques from linguistics, computer science, artificial intelligence, translation theory, and statistics to achieve the desired result [2]. Currently, Machine Translation is one of the most heavily researched areas in Natural Language Processing. Interest in Machine Translation began as early as 1946, when A.D Booth proposed the idea of using electronic computers for Machine Translation [5]. Since then, the area of Machine Translation has made tremendous strides in very less time. Although there are many approaches to Machine Translation, one of the most significant is the use of Statistical Machine Translation. Statistical Machine Translation is an approach to Machine Translation where translation from one language to another is done by analyzing a huge parallel text corpora and then deriving statistical parameters based on which it translates one language to another. Online Machine Translation services like Google, Bing translator all makes use of Statistical Machine Translation to reach the results. Its been noticed that within last few years the use of Assamese language in the web has tremendously increased. But unlike other Indian languages or any other languages which has a presence in the web, there is no translation system that could provide instant and automatic translation to

the users when the need arises. By building a machine translation system that would provide instantaneous translation; will bridge that gap. This paper try to build a system that translates English text to Assamese text by using different statistical parameters. The translation is approached by using Phrase based Statistical Machine Translation, where sentences are divided into small phrases and later depending upon the maximum probable value of each phrase and find the final translated text. Here, it takes bilingual Assamese and English text as the data set and train the data set using statistical parameters to provide automatic translation. The system picks the best translation among the various generated outputs depending on the maximum probability value. This way the best possible English sentence for the desired Assamese sentence is obtained.

## 2. RELATED WORKS AND LITERATURE SURVEY

Statistical translation models have evolved from the word-based models originally proposed by Brown [3] to syntax-based and phrase-based techniques. The beginnings of phrase-based translation can be seen in the alignment template model introduced by Och et al. [18]. Koehn et al. [11] propose certain heuristics to extract phrases that are consistent with bidirectional word-alignments generated by the IBM models [1]. Phrases extracted using these heuristics are also shown to perform better than syntactically motivated phrases, the joint model, and IBM model 4 [13]. Syntax-based models use parse-tree representations of the sentences in the training data to learn, among other things, tree transformation probabilities. These methods require a parser for the target language and, in some cases, the source language too. Yamada and Knight [21] propose a model that transforms target language parse trees to source language strings by applying reordering, insertion, and translation operations at each node of the tree. Wang [20] propose methods based on tree-to-tree mappings. Recently, various pre-processing approaches have been proposed for handling syntax within SMT. These algorithms attempt to reconcile the word order differences between the source and target language sentences by reordering the source language data prior to the SMT training and decoding cycles. Nieben and Ney [17] propose some restructuring steps for German-English SMT. Callison et al. [4] propose German clause restructuring to improve German-English SMT.

## 2.1 Existing Machine Translation Systems

The research and application of Statistical Machine Translation has increased by leaps and bounds in the last two decades. State of the art Statistical Machine Translation Systems are available freely for most of the high resource languages like French, German, and Chinese etc. Owing to the large availability of parallel text from sources like European Union meetings, available of bilingual web sites etc. makes it those systems on par with human translators. The use of Statistical Machine Translation in India has also attracted a high level of interest. Many systems are available for translation of English to Hindi, Urdu, Tamil, Punjabi and Bengali. Here, lets take a look at some of the most widely used Statistical Machine Translation Systems.

- (1) *Google Translate*: Google Translate is a freely available multilingual Statistical Machine Translation System provided by Google Inc. It is one of the most widely used translation services. Though Google Translate applies direct machine translation to most languages, it applies a different approach for languages where direct translation is not possible. In that case, it does not translate from one language to another ( $L1 \rightarrow L2$ ), but often translates first to English and then to the target language ( $L1 \rightarrow EN \rightarrow L2$ ). Currently, Google Translate supports 64 languages.
- (2) *Bing Translator*: Bing Translator which was previously known as Live Search Translator and Windows Live Translator is a user facing translation portal provided by Microsoft as part of its Bing services to translate texts or entire web pages into different languages. All translation pairs are powered by the Microsoft Translator statistical machine translation platform and web service, developed by Microsoft Research, as its back end translation software.
- (3) *Matel*: Matel is a translation system in the late seventies by Siemens-Nixdorf, with the University of Texas. It uses the concept of a controlled language to achieve high quality translation in various technical domains. It produces indicative translation for general texts, which needs to be post-edited for style.
- (4) *Machine translation project in India*: India has 18 constitutional languages, which are written in 10 scripts. Hindi is the official language of the country and English is widely used in the media, commerce, science and technology and education. States like Assam have their own regional languages like Assamese, and others have different languages which can be either Hindi or one of the other constitutional languages. Only about 5% of the population speaks English. There is big market for translation between English and various Indian languages, which task is currently performed manually. Two specific examples of high volume manual translation are- translation of news from English into local languages, translation of annual report of government department and public sector units among, English, Hindi and the local languages. This has geared up the interest of Machine Translation in India. Various machine translation projects related to Indian language are shown below:
  - *Anglabharati*: Anglabharati performs the Machine translation from English to Indian languages, used mostly for Hindi and uses rule-based transfer approach for translation. The system handles ambiguity/complexity by post-editing in case of ambiguity, the system retains all possible ambiguous constructs, and the user has to select the correct choices using a post-editing window to get the correct translation.

- *Anubharati*: Anubharati is a recent project at IIT-Kanpur. It uses example based machine translation for dealing with translation from Hindi to English.
- *Anuvadak English-Hindi software*: Anuvadak is one of the very few private sector efforts in MT in India. They have been working on the software called Anuvadak, which is a general-purpose English-Hindi translation tool that supports post-editing.

## 3. PROPOSED METHODOLOGY

In this system, Statistical Machine Translation is used where it finds the maximum probability of the target sentence by using the probabilities of language model and translation model. The language to be translated is regarded as a source language and the language that the source language is to be translated to as a target language. Here, the aim is to translate English sentence into Assamese sentence. So considering  $e$  to be the English sentence and  $a$  to be the possible Assamese sentence, the most likely Assamese translation will be generated.

$$\hat{e} = \operatorname{argmax}_e P(a|e) \quad (1)$$

Here,  $P(a | e)$  is the probability that given an English sentence  $e$ , an Assamese sentence  $a$  was produced from it.

Now, By Bayes theorem, the probability  $P(a | e)$  can be rewritten as,

$$P(a|e) = \frac{P(a)P(e|a)}{P(e)} \quad (2)$$

So, equation (1) is

$$\hat{e} = \operatorname{argmax}_e \frac{P(a)P(e|a)}{P(e)} \quad (3)$$

Because,  $P(e)$  is independent here, equation (3) can be simplified to obtain the equation,

$$\hat{e} = \operatorname{argmax}_e P(a)P(e|a) \quad (4)$$

So, for translation of the English sentence into Assamese sentence three components are needed:

- (1) *Language Model*: The kind of sentences those are likely to appear in the language ' $a$ '. This is known as the language model  $P(a)$ .
- (2) *Translation Model*: The way sentences in  $e$  get converted to sentences in  $a$ . This is called the translation model  $P(e|a)$ .
- (3) *Decoding*: The decoding step find the best translation by picking up the translation sequence that gives the highest probability.

Here, in figure 1 the Statistical Machine Translation process is shown, which takes in monolingual English text as the language model and Assamese-English bilingual text as the translation model. After that, the decoder gives the best output translation by finding the translation sequence with the highest probability among all the sequences.

### 3.1 Design and Implementation of the Statistical Machine Translation System:

To implement the Statistical Machine Translation System few open source toolkits are used:

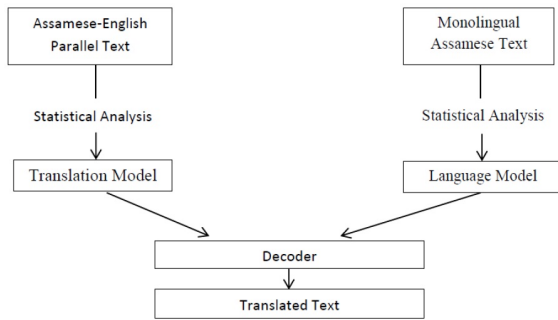


Fig. 1. Statistical Machine Translation Process.

- (1) *Language Modelling*: The language modelling is carried out by an open source toolkit Giza++. GIZA++ implements word alignment model with a dependency of word classes and also implement an alignment model based on Markov model using Forward-Backward algorithm. Giving a bilingual parallel corpus to GIZA++ which in turn uses an unsupervised learning algorithm called Expectation-Maximization algorithm, to compute the translation probabilities.
- (2) *Translation Modeling*: The translation modeling is carried out by using a tool Stanford Research Institute Language Modeling (SRILM). SRILM is used for building and applying various statistical language models. The main objective of SRILM is to support language model estimation and evaluation. Normally SRILM performs the following two important functions:
  - Generation of n-gram counts file from the monolingual target language corpus.
  - Training the language from the n-gram count file.
- (3) *Decoding*: It uses the open source toolkit Moses for the decoding phase. MOSES trains the translation models for any language pair based on the parallel corpus. MOSES is uses phrase based Statistical Machine Translation model for finding the highest probability translation among the exponential number of choices.

### 3.2 BUILDING PARALLEL TEXT CORPUS

A parallel corpus is a collection of text that is paired with its translated text in another language. The first step towards achieving high quality translation in Statistical Machine Translation is the availability of parallel text corpus [19]. It is the most valuable resource for developing Statistical Machine Translation. Parallel text corpus for most languages is freely available as open source for academic pursuits. These texts are in order of millions of words and serve as the foundation for Statistical Machine Translation. These high quality translated text are freely available in the web for instance the Europarl corpus, which contains the translated text of proceedings in the European Union meetings. Many other are also distributed by the Linguistic Data Consortium of the University of Pennsylvania, EMILLE corpus etc.[16]. In India, parallel text corpus for most Indian languages can be obtained from CDAC or Language Development of India Council. The bilingual texts are generally aligned at sentence level; every sentence in one text has a parallel translation in another text. For Statistical Machine Translation from Assamese to English the parallel corpora can be: Currently, the parallel corpus stands at a total of 750 sentences. Although a much larger text corpus is required for building the Sta-

English Corpora	Assamese Corpora
The river Brahmaputra is very wide.	ব্ৰহ্মপুত্ৰ নদী বৰ বহল
It is difficult to cross this river.	এই নদী পাৰ হোৱাটো অসুবিধা
Floods are major problem in Assam.	বানপানী অসমৰ প্ৰধান সমস্যা
The leaves are cut	পাতখিনি কাটি দিয়া
Tea leaves are brought into a room	চাহপাত খিনি কোঠালৈ অনা
The room is very large and cool	কোঠা টো যথেষ্ট আহল-বহল ঠাণ্ডা

Fig. 2. Sample Parallel Corpus.

tistical Machine Translation System, in this phase this system will be tested with this corpus.

### 3.3 LANGUAGE MODELING

It is one of the most important aspect of SMT. Language Modeling calculates the probability  $P(a)$  of occurrence of a particular word after another word in the language  $a$ . Language Modeling is of paramount importance to SMT as this provides the output text to be fluent and adequate. Language Model serves two purposes in SMT.

- (1) *Word Order*: The Language Model gives an estimation which word would follow a sequence of words by giving various probabilistic values to the words. Considering two sentences:
  - $P(LM_1)$ : *I go to school.*
  - $P(LM_2)$ : *go I school to.*
 After analyzing the text corpus, the sequence of words *I go to school* would be higher than the sequence of words *go I to school*. So, a higher probability would be assigned to  $LM_1$  as opposed to  $LM_2$ . Here the probabilistic value of  $LM_1$  would be higher than probabilistic value of  $LM_2$ .  
 $P(\text{go I school to.}) > P(\text{I go to school.})$
- (2) *Word Choice*: Another significant use of Language Model is that it helps in choice of words. Let us consider a word Assamese word "GHAR" . This translates to both the words "house" and "home" in English. Here too, the language model helps to make the correct estimation based on the assigned probability. So considering two sentences,
  - $P(LM_1)$ : *I am going home.*
  - $P(LM_2)$ : *I am going house.*
 Here, the higher probability would be assigned to  $LM_1$  as opposed to  $LM_2$ .  
 $P(\text{I am going home.}) > P(\text{I going house.})$ 
 Language Modeling assigns a probability to a sequence of words  $P(w_1, w_2, \dots, w_n)$  by probability distribution. After analyzing the text corpus, different probabilities are assigned to different words. The Language Model would give a higher preference to correct word order to incorrect word order.
- (3) *N-Grams*: The Language Model used in this methodology is the n-gram language model. In an n-gram model, the probability of a word given all previous words is approximated by the probability of the word given the previous N-1 words. The approximation thus works by putting all contexts that agree in the last N-1 words into one equivalence class. With  $N = 1$  gives the

unigram model, N = 2 gives what is called the bigram model and N = 3 gives the trigram model [6].

n-grams analyses statistical data to estimate how likely words are to follow each other. Taking into context  $LM_1$  : *I go to home* and  $LM_2$  : *I go to house*, while analysing a large amount of text, it is found that the word 'home' followed by 'go to' appears more often than 'house' followed by 'go to'.

The key to collecting the statistical data is analyzing large amount of text and counting how often the word w appears in the corpus. But long sequences of text causes a lot of ambiguity and so the computation of  $P(w)$  has to be broken down into smaller steps. This enables us to collect sufficient statistics and estimate the probability distribution.

In the n-gram model, the process of prediction of word sequence w is broken up into one word at a time. It first breaks down the probability using the Markov Chain Rule.

$$P(w_1, w_2, \dots, w_n) = P(w_1)P(w_2|w_1)\dots P(w_n|w_1, w_2, \dots, w_{n-1}) \quad (5)$$

To count the probabilities, approximately each component is used in the product using the Markov Assumption which gives,

$$P(w_i|w_1w_2\dots w_{i-1}) \approx P(w_i|w_{i-k}\dots w_{i-1}) \quad (6)$$

Markov Assumption is used so that it can limit the number of words which will affect the probability of the upcoming word. After finding the probabilities of each word in correspondence to the previous word, finally, the probability of a sentence which is the product of the conditional probabilities of each word  $w_i$  given by the previous one. So, the language model is,

$$P(w_1w_2\dots w_n) = \prod_{n=1}^n P(w_i|w_{i-(n-i)}, \dots, w_{i-1}) \quad (7)$$

- (4) *Bigram Model*: In the methodology the bigram model is used. The bigram model uses a word to predict the probability of the next word. That is, by counting the probability how often in the training corpus, the word  $w_2$  is followed by the word  $w_1$ , in contrast to other words. The formula for calculating the bigram is:

$$P(w_1|w_2) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)} \quad (8)$$

For example, to count the probability of a sentence, *I want to sleep*, it is split up by bigram count as:

$$P(I \text{ want to sleep}) =$$

$$P(I|START)P(want|i)P(to|want)P(sleep|to)$$

Where START denotes start of a sentence and  $P(I|START)$  is the probability that a sentence starts with the word *i*.

Now, estimating the bigram probabilities of the given sentence in the corpus, giving Biagram count as shown in figure 3.

Taking into account the probabilities, the probability of the sentence is calculated as:

$$P(I \text{ want to sleep}) = 0.24 * 0.09 * 0.16 * 0.08 = .00027$$

Therefore the probability that the sentence "*I want to sleep*" is 0.00027 in the whole corpus.

BIGRAM	PROBABILITY
START i	0.24
I want	0.09
want to	0.16
to sleep	0.08

Fig. 3. Bigram count table.

- (5) *Count Smoothing*: While counting the n-grams, there may be a possibility that the n-gram may not appear in the corpora. This would give the probability of 0 to the corpora. Now, the product of the whole probability would turn into 0 after multiplying the probabilities of the n-grams. To take care this smoothing algorithm is introduced which would take care of such n-grams. To apply count smoothing, add one to the count of all possible sequences. Now, there will be no n-gram with count 0. So, the bigram model is,

$$P(w_1|w_2) = \frac{\text{count}(w_1, w_2) + 1}{\text{count}(w_1) + v} \quad (9)$$

Where, v is the number of word types in the vocabulary.

This ensures that each bigram have at least a count of 1, so sequence that do not occur, will consequently have a non-zero probability.

### 3.4 TRANSLATION MODEL

The translation model generates the target language sentence *a* from the source language sentence *e* by assigning a probability  $P(e|a)$  to the source language sentence and the target language sentence. The main aim of the translation model is to find the text sequences with the highest probability of lexical correspondence.

- (1) *Phrase Based Translation Model*: The Phrase based Translation Model performs the machine translation by dividing a sentence into number of smaller phrases and then translating each small word sequence one at a time [12, 14].

In the first phase of the translation model, the parallel text corpora is analyzed and statics are gathered. Let takes an example of an Assamese phrases "*AaiBudhi*". So by analyzing a large parallel text collection statistics can be gathered. Taking this statistics into account, the lexical translation probability distribution table can be formed from this data, which helps to compute most likely English translation phrase for the Assamese word. So, the phrase table may be formed as shown in figure 4:

English	$P(a   e)$
this idea	0.6227
the idea	0.1068
an idea	0.0341
This plan	0.0227
This concept	0.0159
it	0.069

Fig. 4. Phrase Translation Table

- (2) *Obtaining the maximum probable value in the translation model:* The goal is to obtain the maximum value of the probability distribution table so to get the best Assamese translation  $e_{max}$  for the English sentence.

$$\begin{aligned} e_{max} &= \operatorname{argmax}_e P(a|e) \\ &= \operatorname{argmax}_e P(e|a)P(e) \end{aligned} \quad (10)$$

Now, taking the English sentence  $e$ , and decompose it into  $n$  phrases  $\bar{e}_n$  and consequently it is translated into the Assamese phrase  $\bar{a}_n$ , and multiply the probabilities as shown,

$$P(\bar{a}_n|\bar{e}_n) = \prod_{n=1}^n P(\bar{e}_n|\bar{a}_n)d(\text{start}_i - \text{end}_{i-1} - 1) \quad (11)$$

Here, considering the phrase reordering distance which is the distortion caused during the reordering of the phrases. It gives the number of words having a different position in the Assamese sentence than it had in the English sentence. The distortion here is parameterized by,

$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

where  $\text{start}_i$  is the position of the first word of the Assamese phrase that translates to the  $i^{\text{th}}$  English phrase, and  $\text{end}_i$  as the position of the last word of that Assamese phrase.

Let us consider an example,

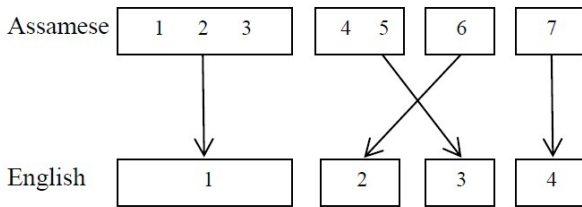


Fig. 5. Distance Reordering

So, the distance reordering table is as Table 1.

Table 1. Distance Reordering Table

Phrase	Translates	Movement	Distance
1	1-3	Start at Beginning	0
2	6	Skip over 4-5	+2
3	4-5	Move Back over 4-6	-3
4	7	kip over 6	+1

So after, getting the maximum value for different phrases, the phrases are arranged in the phrase table along with the probability values.

### 3.5 Decoding

Decoding process is the process of finding the best possible probability in the search space of all possible translation [8]. The decoder takes in the output from the Language Model and the Translation model and looks for the most probable translation. Exhaustively examining all possible translations, scoring them and picking the best is computationally expensive. The decoder uses a heuristic search method to find the

best possible translation [7,9].

- (1) *Decoding Process:* The decoding process starts by trying to find the best probability value for the given formula:

$$\hat{e} = \operatorname{argmax}_e \prod_{n=1}^n P(\hat{e}_n|\hat{a}_n)d(\text{start}_i - \text{end}_{i-1} - 1)P(a) \quad (12)$$

Several components contribute to the overall score of finding the best probability which includes:

- Phrase translation probability  $P(\hat{e}_n|\hat{a}_n)$ .
- The reordering model  $d$ .
- The language model  $P(a)$ .

Given all  $n = 1, 2, \dots, n$  input phrases  $P(\hat{e}_n)$  and output phrases  $P(\hat{a}_n)$ , the positions  $\text{start}_i$  and  $\text{end}_i$ , it is straightforward to compute the probability of the translation. Moreover, during the translation process, it is able to compute partial scores for the partial translations that have been constructed. So, instead of computing the translation probability for the whole translation only when it is completed, it computes incrementally and add to the score as it is constructed.

Adding one phrase translation every time factors the scores from the three model components in the formula :

- *Translation model:* When it picks an input phrase  $P(\hat{e}_n)$  to be translated as an output phrase  $P(\hat{a}_n)$ , it consults the phrase translation table to look up the translation probability for this phrase pair.

- *Reordering model:* For reordering, it has to keep track of the end position of the input of the previous phrase it has translated,  $\text{end}_{i-1}$ . It also knows where it is picking the current phrase and hence has a start position in the input,  $\text{start}_i$ . Armed with these two numbers, it can consult the distortion probability distribution  $d$  to get the reordering cost.

- *Language model:* For the application of language models to translation, that once it adds new words to the end of the sequence, it can compute their language model impact by consulting the language model  $P(a)$ . For this it needs to keep track of the last  $n - 1$  words it added, since they form the history of the language model.

Taking into account of the three model components i.e. phrase translation, reordering, and language model, whenever it adds a new translated phrase to the partially constructed translation, it needs to know the identity of the phrase pair, location in the input, and some information about the sequence that have been constructed so far.

## 4. EVALUATION

Statistical Machine Translation usually requires a huge text corpus for analyzing the statistical parameters and to give a good result [10]. The System is built with a corpus of about 5000 sentences in bilingual text corpus. The system is tested with a small set of corpus. The system is tested mainly with some simple sentences as text corpus was very small and the range of text contained in it was not much. The outputs were as follows in figure 6:

## 5. CONCLUSION AND FUTURE SCOPE

This paper presented an approach of building a Statistical Machine Translation System that would give translated text from English to

English	Assamese
My name is Ram	মোৰ নাম Ram
I am a student	মই এজন ছাত্ৰ
He is a good boy	সি এজন ভাল ল'ৰা
I am hungry	মোৰ ভোক লাগিছে
Come	আহা

Fig. 6. Translation Results.

Assamese. The Statistical Machine Translation System took a set of bilingual text corpus of Assamese and English languages as input data. It used various statistical parameters to compute the Assamese text sequence which had the maximum probability value of being translated from the corresponding English language. The system is tested with a text corpus of about 5000 sentences and received satisfactory results.

The results were satisfactory, owing to the constraint that the data set was small as compared to other language pairs. Typically, in Statistical Machine Translation Systems, the larger the text corpus, the better is the result as more statistics can be gathered from a larger data set. There is scope for adding text to the corpus that have been built, so over time, new data can be incorporated, which will improve the statistics gathered for the system.

## 6. REFERENCES

- [1] Al-Onaizan, Y. et-al, "Distortion models for statistical machine translation", In Proceedings of ACL-COLING, July 2006, pp. 529 – 536.
- [2] Birch, A. et-al, "Constraining the phrase-based joint probability statistical translation model", In Proceedings of HLT-NAACL Workshop on Statistical Machine Translation, April 2006, pp 154 – 157.
- [3] Brown, P.F. et-al, "The mathematics of statistical machine translation: Parameter estimation"; Computer Linguist., vol.19, no.2, pp. 263 – 311.
- [4] Callison, C. et-al, "Scaling phrase-based statistical machine translation to larger corpora and longer phrases"; In Proceedings of the Association for Computational Linguistics (ACL), June 2005, pp. 255 – 262.
- [5] Johnson, H. et-al, "Improving translation quality by discarding most of the phrase table; In Proceedings of EMNLP-CoNLL", June 2007, pp. 967 – 975.
- [6] Jurafsky and Martin, "Speech and Language Processing: An Introduction to Natural Language Processing", Computational Linguistics, and Speech Recognition, 2nd Ed. Prentice-Hall, pp. 232 – 241.
- [7] Knight, K., "Automating knowledge acquisition for machine translation", AI Magazine., November 2007, vol.18, no. 4, pp. 81 – 96
- [8] Knight, K., "Decoding complexity in word-replacement translation models", Computer Linguist., April 1999, Vol. 25, no. 4, pp. 607 – 615
- [9] Koehn, P., "Pharaoh: A beam search decoder for phrase-based statistical machine translation models" In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), April 2004.
- [10] Koehn, P., "Statistical significance tests for machine translation evaluation", In Proceedings of EMNLP., December 2004, pp.388-395.
- [11] Koehn, P., "Statistical Machine Translation", 1st Ed., Cambridge University Press., pp. 314 – 352
- [12] Koehn, P. et-al, "Factored translation models", In Proceedings of EMNLP-CoNLL., June 2009, pp. 868 – 876
- [13] Koehn, P. et-al, "Statistical phrase-based translation", In Proceedings of HLTNAACL., May 2003, pp. 127 – 133
- [14] Lopez, A., "Hierarchical phrase-based translation with suffix arrays", In Proceedings of EMNLP CoNLL., June 2007, pp.976 – 985
- [15] Lopez, A. Et-al, "Improved HMM alignment models for languages with scarce resources", In Proceedings of ACL Workshop on Building and Using Parallel Texts., April 2005, pp. 83 – 86.
- [16] Manning and Schutze, "Foundations of Statistical Natural Language Processing", MIT Press, pp.89 – 94.
- [17] Nieben, S., et-al, "Statistical machine translation with scarce resources using morpho-syntactic information", Comput. Linguist., May 2004, vol. 30, no. 2, pp. 182 – 204.
- [18] Och, F. J., "An efficient method for determining bilingual word classes", In Proceedings of the European Chapter of the Association for Computational Linguistics (EACL.), March 1999, pp. 71 – 76.
- [19] Resnik, P. et-al, "The Web as parallel corpus", Comput. Linguist. July 2003, vol.29, no. 3, pp. 349 – 380.
- [20] Wang, Y. Et-al, "Decoding algorithm in statistical machine translation", In Proceedings of ACL-EACL., January 1997, pp. 366 – 372.
- [21] Yamada, K. Et-al, "A decoder for syntax-based statistical MT", In Proceedings of the Association for Computational Linguistics (ACL.), June 2002, pp. 303 – 310.