I.J. Modern Education and Computer Science, **2014, 7, 1-8**
Published Online July 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2014.07.01

A Pure EBMT Approach for English to Hindi Sentence Translation System

Ruchika A. Sinhal
RCOEM, Department of CSE, Nagpur, India
Email: ruchisinhal04@gmail.com

Kapil O. Gupta
DMIETR, Department of IT, Wardha, India
Email: kaps04gupta@gmail.com

***Abstract*—**The paper focuses on Example Based Machine Translation (EBMT) system that translates sentences from English to Hindi. It uses the parallel corpus for translating sentences. Development of a machine translation (MT) system typically demands a large volume of computational resources. Requirement of computational resources (for example, rules) is much less in respect of EBMT. This makes development of EBMT systems for English to Hindi translation feasible, where availability of large-scale computational resources is still scarce. Example based machine translation relies on the database for its translation. The frequency of word occurrence is important for translation in EBMT in the following research.

***Index Terms*—**Example Based Machine Translation, Parallel Corpus, Word Matching, Data Dictionary, Matrix.

I. Introduction

A. Machine Translation

The natural language is used by every common man. The language which human beings speak is termed as natural language. People use natural language for communication. Natural language processing includes refining, modifying and translating i.e. operating on one of these natural languages.

When all humans are able to speak, then what is the need for processing this natural language in our life? The need can be explained by divergence. The divergence is difference in language or the form of text in which the language is present. In India itself there are more than 20 languages spoken. The most ancient of all languages is Sanskrit. Many people do not understand Sanskrit but they can if the text is translated into their national language or languages they are familiar with. Therefore, for understanding and making communication easy there is a basic need of translator. This translation can be done by humans; so why there is a need of machine translation? The first reason is that the “world of text” is huge. There are many large documents to be translated and it is not possible for a human to translate gigabytes of data in a short time. To reduce the human efforts and to give the results quickly the machine translators are used which can translate the text from one language to another by just one click. A second reason is that the all technical materials are too boring for human translators to translate as humans do not like to translate them continuously. Hence they look for help from computers. Thirdly, as far as large corporations are concerned, there is the major requirement that terminology is used consistently, the terms to be translated in the same way every time. Computers are consistent, but human translators tend to seek variety; they do not like to repeat the same translation and this is not good for technical translation. A fourth reason is that the use of computer-based translation tools can increase the volume and speed of translation throughput, and organizations like to have translations immediately. The fifth reason is that top quality human translation is not always needed. Computers do not produce good translations. The fact is that there are many different circumstances in which top quality translation is not essential, and in this case, automatic translation can be used widely [1].

The need for machine translation can be briefly stated into following points briefly:

- Too much to be translated
- Boring for human translators
- Major requirement that terminology used consistently
- Increase speed and throughput
- Top quality translation not always needed
- Reduced cost

B. History of Machine Translation

W. John Hutchins, 1986 explained very vast history of machine translation [1-3]. Many are under the impression that MT is something quite new. It has a long history – almost since before electronic digital computers existed. In 1947 when the first non-military computers have been developed, the idea of using a computer to translate has been proposed. In July 1949 Warren Weaver [4] (a director at the Rockefeller Foundation, New York) proposed method which introduced Americans to the idea of using computers for translation. From this time on, the idea spread quickly, and in fact machine translation has been to become the first non-numerical application of computers. The first conference on MT was held in 1952 [5]. Just two years later, there has been the first demonstration of a translation system in January 1954 [6]. Unfortunately it has been the wrong kind of attention as many readers thought that machine translation has been just around the corner and that not only would translators be out of a job but everybody would be able to translate everything and anything at the touch of a button. It gave quite a false impression. However, it has been not too long before the first systems have been in operation, even though the quality of their output has been quite poor. In 1959 a system has been installed by IBM at the Foreign Technology Division of the US Air Force [7], and in 1963 and 1964 Georgetown University, one of the largest research projects at the time, installed systems at Euratom and at the US Atomic Energy Agency. But in 1966 there appeared a rather damning report for MT from a committee set up by most of the major sponsors of MT research in the United States. It found that the results being produced have been just too poor to justify the continuation of governmental support and it recommended the end of MT research in the USA altogether. It advocated the development of computer aids for translators. Consequently, most of the US projects – the main ones in the world at that time – came to an end. The Russians, who had also started to do MT research in the mid 1950's, concluded that if the Americans were not going to do it any more than they would not either, because their computers have not been as powerful as the American ones. However, MT did continue in fact, and in 1970 the Systran system has been installed at the US Air Force (replacing the old IBM system), and that system for Russian to English translation continues in use to this day [8]. The year 1976 is one of the turning points for MT. In this year, the Météo system for translating weather forecasts has been installed in Canada and became the first general public use of a MT system [9]. The European Commission decided to purchase the Systran system and from that date its translation service has been developed and installed versions for a large number of language pairs for use within the Commission. Subsequently, the Commission decided to support the development of a system designed to be better than Systran, which at that time has been producing poor quality output, and began support for the Eurotra project – which did not produce a system in the end. During the 1970's other systems began to be installed in large corporations [10]. In 1981, came the first translation software for the newly introduced personal computers, and gradually MT came into more widespread use [11]. In the 1980's there had been a revival of research, Japanese companies began the production of commercial systems, and computerized translation aids became more familiar to professional translators. Then in 1990, the first translator workstations came to the market [12]. In the last decade MT has become an online service on the Internet [13-14].

The term machine translation (MT) is translation of one language to another. The ideal aim of machine translation system is to produce the best possible translation without human assistance. Basically every machine translation system requires automated programs for translation, dictionaries and grammars to support translation [15].

The work in this paper is organized in five parts. First part discusses the review of literature on machine translators and the related work in development of example based machine translators. The further discussion focuses on the proposed methodology and the algorithm implemented in translation and training of the system. The implementation and working of the system is discussed later followed by drawn conclusion, result and the future scope for the translation system.

II. Literature Review

A. *Example Based Machine Translation*

EBMT is a corpus based machine translation, which requires parallel-aligned three machine-readable corpora. Here, the already translated example serves as knowledge to the system. This approach derives the information from the corpora for analysis, transfer and generation of translation. These systems take the source text and find the most analogous examples from the source examples in the corpora. The next step is to retrieve corresponding translations. And the final step is to recombine the retrieved translations into the final translation.

EBMT is best suited for sub-language phenomena like – phrasal verbs; weather forecasting, technical manuals, air travel queries, appointment scheduling, etc. Since, building a generalized corpus is a difficult task, the translation work requires annotated corpus, and annotating the corpus in general is a very complicated task.

Nagao (1984) has been the first to introduce the idea of translation by analogy and claimed that the linguistic data are more reliable than linguistic theories [16]. In EBMT, instead of using explicit mapping rules for translating sentences from one language to another, the translation process is basically a procedure for matching the input sentence against the stored translated examples. Fig. 1 shows the architecture of a pure EBMT [17].

The basic tasks of an EBMT system are –

- Building Parallel Corpora
- Matching and Retrieval
- Adaptation and Recombination

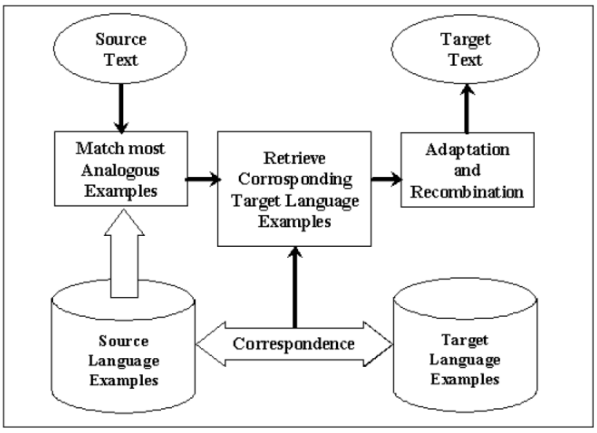

Source Text
Target Text
Match most Analogous Examples
Retrieve Corresponding Target Language Examples
Adaptation and Recombination
Source Language Examples
Correspondence
Target Language Examples

Fig. 1 EBMT Architecture

The knowledge base, parallel aligned corpora consist of two sections, one for the source language examples and the other for the target language examples. Each example in the source section has one to one mapping in the target language section. The corpus may be annotated in accordance with the domain. The annotation may be semantic (like name, place and organization) or syntactic (like noun, verb, preposition) or both. For example, in the case of phrasal verb as the sub-language the annotations could be subject, object, preposition and indirect object governed by the preposition. In the matching and retrieving phase, the input text is parsed into segments of certain granularity. Each segment of the input text is matched with the segments from the source section of the corpora at the same level of granularity. The matching process may be syntactic or semantic level or both, depending upon the domain. On syntactic level, matching can be done by the structural matching of the phrase or the sentence. In semantic matching, the semantic distance is found out between the phrases and the words. The semantic distance can be calculated by using a hierarchy of terms and concepts, as in WordNet. The corresponding translated segments of the target language are retrieved from the second section of the corpora. In the final phase of translation, the retrieved target segments are adapted and recombined to obtain the translation. It identifies the discrepancy between the retrieved target segments with the input sentences' tense, voice, gender, etc. The divergence is removed from the retrieved segments by adapting the segments according to the input sentence's features.

B. Machine Translators

The table 1, describes the Example Based Machine Translator Implementation. These translators have different domain of their working. These translators used different computational resources for their operation.

Table 1: Example Based Machine Translators

Sr. No.	Name of Translator	Languages Used	Year
1	Anubharti	Hindi- Indian Language	1995
2	Vaasaanubaada	Bengali-Assamese	2002
3	Anubharti-II	Hindi- Indian Language	2004
4	Shiva	Hindi- Indian Language	2004
5	Anubad	English-Bengali	2004
6	IBM-MTS	Eng-Hindi	2006

- Anubharti-II Technology

R.M.K Sinha in 2004, has proposed a system with an approach for machine aided translation having the combination of example-based and corpus based approaches and some elementary grammatical analysis. In ANUBHARTI the traditional EBMT approach has been modified to reduce the requirement of a large example base. ANUBHARTI-II uses Hindi as a source language for translation to other Indian language [18].

- VAASAANUBAADA A Bilingual Bengali

Vijayanand Kommaluri, Sirajul Islam Choudhury and Pranab Ratna in 2002 proposed Assamese automatic MT system for translating the news texts by using the Example Based Machine Translation (EBMT) technique. In this the translation is done at sentence level. Some preprocessing and post processing work has to be done for the translation. The longer sentences were fragmented at punctuation, which gives high quality translations. Backtracking is used when exact match does not occur at the sentence level, which results in further fragmentation of the sentence [19].

- Shiva and Shakti

In 2004 two MT systems were developed jointly by IIIT Hyderabad, IIS Bangalore and Carnegie Mellon University USA for translation from English to Hindi. The Shiva by Sivaji Bandyopadhyay [20-21], system uses example based approach and the Shakti by R. Moona Bharati, P. Reddy, B. Sankar, D.M. Sharma, R. Sangal [20], system uses rule-based approach with statistical approach for MT. The Shakti system is working for three target languages Hindi, Marathi and Telugu.

- ANUBAD Hybrid Machine Translation System

Sivaji Bandyopadhyay in 2004, proposed a MT system for translating English news headlines to Bengali at Jadavpur University Kolkata [22]. Saha Gautam Kumar in 2005 developed EB-ANUBAD [23], system for translating English to Bengali language and it showed 98% correct results although the output has been in English language. The approach taken for the translation has been the hybrid approach of rule based and transfer based with a parser for both morphological as well as for the Lexical parsing of the text. A brief description of the various machine translation systems is given above in the table 1.

- IBM-English-Hindi Machine Translation System

D. Gupta, N. Chatterjee [24] and Raghavendra Udupa, Tanveer A. Faruquie [25] in 2006 proposed to develop a MT based on Example Based approach and later on shifted to the Statistical approach for machine translation from English to Indian Languages in IBM India Research Lab.

- English to {Hindi, Kannada, and Tamil} and Kannada to Tamil language pair EBMT

B. K.Murthy, W. R.Deshpande in 2006 [24, 26] developed a MT system based on bilingual dictionary of sentences, phrases, words and phonetic dictionary. Each dictionary contains parallel corpora for the language pair based on EBMT. EBMT has a set of 75000 commonly used sentences from English and translated into the target Indian Languages.

III. Proposed Methodology

There are two objectives in the presented research. Proposed algorithms for the two objectives are explained below in stepwise manner. The training corpus is the parallel database containing 677 sentences. The corpus generated is not preprocessed. The examples contained in training corpus are newspaper headlines.

The database trained forms the matrix. This matrix can be stored in mat file for later use. Once the system is trained, there is no need to train the system later till there is some modification made in the database.

A. Training Algorithm

The first objective for the system designed is to train the system for translation. The training algorithm stated below is designed to create matrix data structures later used in the translation phase.

```
Step 1: Initialize the Similarity Matrix, Training Matrix
        and TagMatrix as zero.
Step 2: Read first sentence form database.
Step 3: Tokenize the string.
Step 4: Read sentence from database to compare with the
        first.
Step 5: Tokenize the string.
Step 6: Compare two tokens if match found, then update
        similarity matrix and increment counter for word
        else compare for next word.
Step 7: Store the counter for the training matrix.
Step 8: Check current variable if less than size of
        database, then increment counter and go to Step 4.
Step 9: Check current variable if less than size of
        database, then increment counter and go to Step 2
        to reinitialize the counter for the occurrence of
        word.
```

The training module forms the matrix; this matrix is used later for matching in translation phase. The indexes of sentence containing the corresponding word are matched to find the common Hindi string which is combined to form the output. The training matrix gives the number of occurrence of the word in the corpus. The system uses these databases as input in translation phase. These data structures are loaded each time.

B. Translation Algorithm

The algorithm for translation of sentence is presented below. The translator performs word based translation.

```
Step 1: Train the database for finding the matrices.
Step 2: Read the input string.
Step 3: Divide the string into tokens.
Step 4: Parse the database for the first word i.e. token.
Step 5: If the word is present only once in database, use
        the dictionary to search for the translation else go
        to Step 6.
Step 6: If the word is present twice in database, then it
        finds the common string in the Hindi language to
        find the match else go to Step 7.
Step 7: The corresponding Hindi sentences are taken and
        the intersection of sentences is found and the
        common string is found.
```

The sentence entered by user is broken into tokens using user defined function. The sentence indexes for these tokens are matched for translation one-by-one from the corpus by user defined String Cmp function. The matched output is then combined to generate the output translated string. The input sentence is not preprocessed to remove stop words e.g. the, is, as etc.

IV. Implementation

The research mainly has two objectives training and translation. Training of database train the database for translation and translator converts the sentence from English to Hindi. These two objectives are explained more in detail below.

A. Training

The training system objective further is divided in three modules. These modules develop the data structures while training which have been used in later objective. The training is performed using the parallel corpus. The parallel corpus containing English-Hindi sentence pair is generated. These modules are described in following parts:-

- Development of Similarity Matrix

The sentence is correct with meaningful alignment of the words. The corpus used for training is made of 677 parallel English-Hindi sentence pairs. Similarity matrix is generated using training corpus to find the existence of each word from sentence in the corpus. Each word is checked for its match in entire corpus of English sentences. If the match is found, the value in the matrix is changed to a non-zero value i.e. 1. The non-zero value indicates the presence of a word in database. If the word is present in database the system compute the matrix by placing 1 in place of existence of word. This check is performed for each sentence and each word present in the corpus. The Similarity Matrix is then stored in .mat format for later use. The values in table 2 show the similarity matrix for first 10 sentences in the corpus. This similarity matrix is calculated for ten English sentences in the corpus.

Table 2: Similarity Matrix for 10 sentences

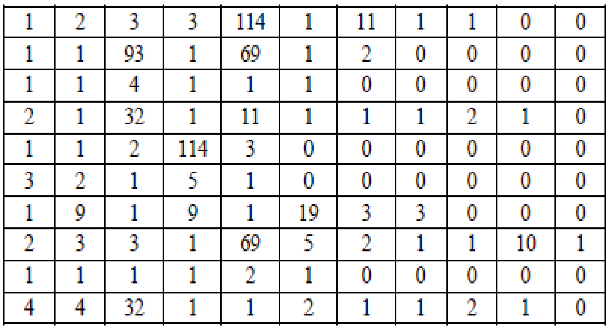
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1

- Development of training Matrix

The training matrix is used to find the occurrence of the word in the entire corpus i.e. the count of the each word in the database. The occurrence gives the frequency of word in the database. The word present in the corpus and its corresponding match can be retrieved using translation algorithm. If the frequency of the word is more, then the accuracy of translation for that word will be more.

The values in table 3 are the occurrence for the corresponding word in the sentence.

Table 3: Training Matrix for 10 sentences



1	2	3	3	114	1	11	1	1	0	0
1	1	93	1	69	1	2	0	0	0	0
1	1	4	1	1	1	0	0	0	0	0
2	1	32	1	11	1	1	1	2	1	0
1	1	2	114	3	0	0	0	0	0	0
3	2	1	5	1	0	0	0	0	0	0
1	9	1	9	1	19	3	3	0	0	0
2	3	3	1	69	5	2	1	1	10	1
1	1	1	1	2	1	0	0	0	0	0
4	4	32	1	1	2	1	1	2	1	0

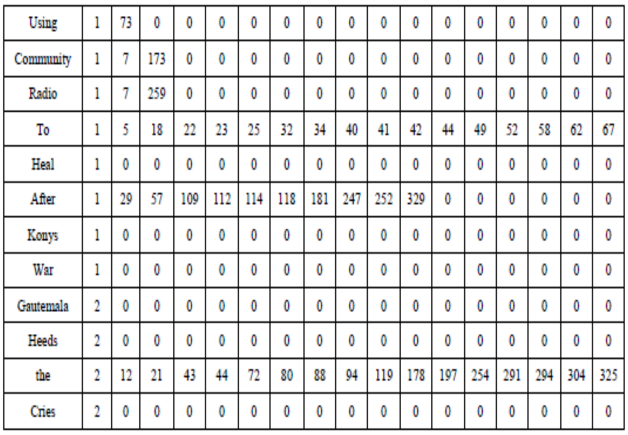
The occurrence of the first word for example "Uganda" is 1, similarly for second word "using" is 2. The index of the occurrence in which sentence is found from the Tagmatrix is explained below.

- Development of Tagging Matrix (TagM)

The tagmatrix stores the sentence number of the occurrence of the word. The sentences are then compared and the intersection is found for the comparison. The common string for the tag matrix is found for the output.

The tag matrix for the first 12 words is shown in the table 4 below the word for corresponding tag is shown in first column. The sentence index is stored in preceding columns. The matching is found with the match using the index of the sentences.

Table 4: Tag Matrix for 12 words



Using	1	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Community	1	7	173	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Radio	1	7	259	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To	1	5	18	22	23	25	32	34	40	41	42	44	49	52	58	62	67
Heal	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
After	1	29	57	109	112	114	118	181	247	252	329	0	0	0	0	0	0
Konys	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
War	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gautemala	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Heeds	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
the	2	12	21	43	44	72	80	88	94	119	178	197	254	291	294	304	325
Cries	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B. Translation of Sentence

This objective translates the sentence from English to Hindi. Word based translation is performed by comparing tokens of the input sentences. The input sentence is divided in word strings called tokens. The space is used as delimiter. The English sentence has SVO sentence structure and Hindi has SOV structure. The correct translation is formed by alignment of text in proper structured format i.e. if sentence is translated from English-Hindi the Subject-Verb-Object is aligned as Subject-Object-Verb in translated output. The translator presented in research-work does not align the text in SOV format for generated Hindi translation. Let us summarize the modules in detail.

- Finding the occurrence of word in database

The word to be translated is checked for its occurrence in the corpus. The occurrence of the word is found from the training matrix generated in training phase. TagM matrix gives the indexes of the English sentence in which the word to be translated is present. These indexes are used to retrieve the corresponding Hindi sentences from the corpus. These Hindi sentences are compared to find the common string. The string common in the sentences is the corresponding translation for the word.

- Matching the word in sentence

The matching of a word in sentence depends on occurrence of the word in the corpus. The retrieved Hindi sentences are tokenized before matching. The matching for the word has been implemented in three cases.

a. Word occurs only once in database

The occurrence of the word is found from the training matrix. If the word exists only once in the database, then comparison of two Hindi sentences is not possible. In such case the dictionary is used. The dictionary containing one-to-one translation of English-Hindi words is used. The word is checked in English words corpus. If the word is present, its corresponding match is extracted from the Hindi words corpus.

b. Word occurs twice in database

The word if present in two English sentences, the corresponding Hindi sentence is retrieved from the corpus. These Hindi strings are passed into user defined function StringCmp( ) which finds the common string comparing both the Hindi sentences.

c. Word occurs thrice in database

The word if present in more than two sentences that is, if its frequency is greater than or equal to three the sentences, it is intersected using intersect( ) the inbuilt Matlab function. The Hindi sentences are passed as input to the function and the common output string is found. The example explained below describes the processing in detail.

For example, the sentence “Kerala state beautiful” is to be translated using the translator, it performs the following steps while translation. The tokens are generated from the input sentence. The sentence is divided in tokens considering space as the delimiter. Therefore the sentence tokens are “Kerala”, “state” and “beautiful”. In the next step, each token is to be processed for translation one-by-one. The occurrence of these tokens is checked from training matrix. The sentences containing these words are found from the tagmatrix. Through the tag of sentences in tagmatrix the corresponding translation is searched for match and translation is extracted from the corpus. Thus the output retrieved is word to word translation.

The example is elaborated in following description.

The text to be compared is “Kerala”. It is compared in the corpus and the tag matrix the tag matrix is displayed below.

TagM = 255 259

i.e. word kerala is present in sentence 255 and 259. These two strings are compared using a Stringcmp function.

S1= 'bljkbyh' ';qxy' 'fuokZlu' '%' 'dsjy' 'mPp' 'U;k;ky;' 'vnkyr' 'us' 'yxkbZ' 'jksd'
S2= 'dsjy' 'esa' 'vkjih,Q' 'la[;k' 'Åij' 'tk' 'ldrh' 'gS'

The common string of the two sentences S1 and S2 is found which is ‘dsjy’ i.e. translation for “Kerala”. The same procedure is followed for the rest of the tokens in the sentences. If the word occurs in matrix three times, the intersection of sentences found from the tag matrix is found for example “state”. The tag matrix for state is:

TagM= 14 98 246

i.e. word state is in sentence 14, 98 and 246. These three strings are intersected using intersect function in MATLAB as,

S1= 'MkDVj' 'gMrky' 'ij' '%' 'ekSr' 'okyksa' 'dh' 'Vksy' 'la[;k' 'jkT;' 'ds' 'vLirkyks' 'es' 'cMh'
S2= 'jkT;' 'ljdkj' 'dks' 'vnkyr' '%' 'D;k' 'flaxwj' ',DV' 'vko’;d' 'Fkk\\'
S3= 'vtsZaVhuk' 'ihfMr' 'jkT;' 'vkradokn' 'ds' 'muds' 'ne' 'ij' 'vc' 'fldkj'

The intersection of these three sentences is ‘jkT;’ which is concatenated with the previous matched output. Thus the sentences index is used for finding the common string to find the right translation.

V. Result Analysis & Discussion

The non parallel testing database of 150 English sentences is created to test the precision and word-strength of translator. The precision is the mathematical measure which gives the quantity up to which the output generated is correct. Word-strength gives the mathematical quantity for the exact word translation for the keywords in the sentence.

Precision is the measure to check the quality of translation. It is given by the following mathematical formula [27]. The precision for the translator presented in the research work is computed by varying the size of corpus used for translation. The table 5 below shows the precision and word-strength of the output generated through the translator. The quantities make it clear as the size of database increases, the precision of the system also increases. The graph is drawn for the tabulated quantities in Fig. 2.

The formula for precision is:-

$$\text{Precision}=\frac{Correct\ Translation}{Correct\ Translation+Unexpected\ Translation} \quad (1)$$

Precision for 10 sentences = 3 / (3+7)
= 0.3

Precision for 50 sentences = 6 / (6+4)
= 0.6

Precision for 677 sentences = 126 / (126+20)
= 0.86

Table 5: Figures Showing Precision for Translation of the System

No. of Sentences used	No. of Sentences tested	Correct Translation	Unexpected Translation	Precision in %
10	10	3	7	30%
50	10	6	4	60%
677	146	126	20	86%

The graph for the translation precision is drawn below in Fig. 2:

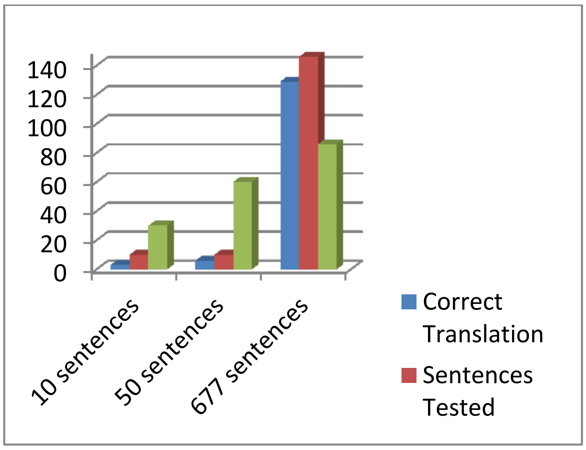


Fig. 2 Graph showing the precision of translator

The database on which testing is performed is also checked for the word strength for the translation generated by the research. The table below gives the details of the word strength in detail.

Word Strength= Correct Word Translation / (Correct Word Translated + Missing Word Translated) (2)

Word Strength for 10 sentences = 20/ (20+7)
= 0.74

Word Strength for 50 sentences = 34/ (34+2)
= 0.94

Word Strength 677 sentences = 489/ (489+20)
= 0.96

Table 6: Figures Showing Word Strength for Translation of the System

No. of Sentences used	No. of Sentences tested	Correct Word Translated	Missing Words Translated	Word Strength in %
10	10	20	7	74.07
50	10	34	2	94.44
677	146	489	20	96.07

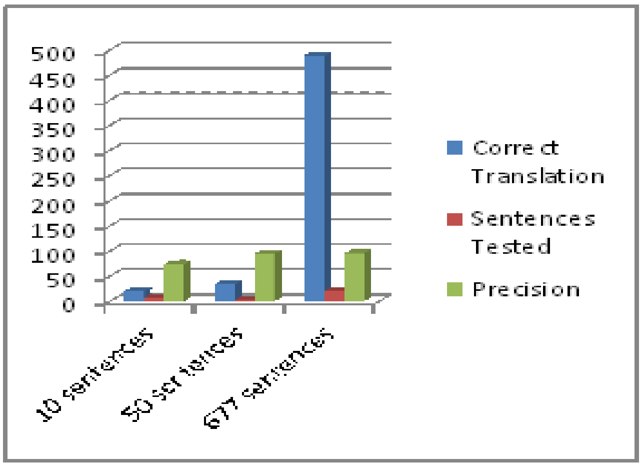


Fig. 3 Graph showing the word strength of translator

VI. Conclusion and Future Scope

This research focuses on simple way of comparing sentence to extract the translation. The research concludes the translator gives the proper expected output to some extent by comparing sentences. The research can be taken to next level by using preprocessed database. The translator presented in the research work does not structure the output sentence in SOV form. Thus the alignment of translation can be done. There is no approach to reduce divergence in generated output, therefore algorithm can be written for reducing the divergence in future.

References

[1] Hutchins W. John and Harold L. Somers, (1992). An Introduction to Machine Translation. *London: Academic Press.*

[2] D. Arnold, L. Balkan, S. Meijer, L.L. Humphreys, L. Sadler: *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, Great Britain, 1994.

[3] Hutchins 95 J. Hutchins: Reflections on the history and present state of machine translation. In Proc. of *Machine Translation Summit V*, pp. 89–96, Luxembourg, July 1995.

[4] W.Weaver. Translation. In W.N. Locke, A.D. Booth, editors, *Machine Translation of Languages: fourteen essays*, pp. 15–23. MIT Press, Cambridge, MA, 1955.

[5] John Hutchins, *Milestones in machine translation No.4: The first machine translation conference*, June 1952 Language Today, no. 13, October 1998, pp.12-13 [Online].

Available: http://www.hutchinsweb.me.uk/Milestones-4.pdf.

[6] The first public demonstration of machine translation: The Georgetown-IBM system, 7th January 1954. [Online]. Available: http://www.hutchinsweb.me.uk/GU-IBM-2005.pdf.

[7] Masterman, Margaret and Kay, Martin, “Operational system (IBM-USAF Translator Mark I), at Foreign Technology Division, USAF, in 1959”. [Online]. Available: www.hutchinsweb.me.uk/sources/Russian-IBM-1959.doc.

[8] Systran, [Online]. Available: http://www.hutchinsweb.me.uk/IntroMT-10.pdf.

[9] The EUROTRA project, [Online]. Available: http://www-sk.let.uu.nl/stt/eurotra.html.

[10] J. Chandioux, A. Grimaila: Specialized machine translation. *In 2nd Conf. of the Association for Machine Translation in the Americas (AMTA 96)*, pp. 206–212, Montreal, Canada, Oct. 1996.

[11] “Machine Translation”, [Online] Available: http://en.wikipedia.org/wiki/Machine_translation

[12] John Hutchins, “The origins of the translator’s workstation”, Machine Translation, vol.13, no.4 (1998), p. 287-307.

[13] Hutchins and Lovtsky, *in press*.

[14] Hutchins, J. 1986. Machine Translation: Past, Present, Future, Ellis Horwood/Wiley, Chichester/New York.

[15] Sergei Nirenburg and Yorick Wilks, Machine Translation.

[16] Makoto Nagao, A Framework of A Mechanical Translation between Japanese and English by Analogy Principle, *In Artificial and Human Intelligence* 1984 [Online]. Available: http://www.mt-archive.info/Nagao-1984.pdf.

[17] Indranil Saha et.al. (2004). Example-Based Technique for Disambiguating Phrasal Verbs in English to Hindi Translation. Technical Report KBCS Division CDAC Mumbai.

[18] R.M.K. Sinha,"An Engineering Perspective of Machine Translation", AnglaBharti-II and AnuBharti-II Architectures. *In proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS- 2004).* November 17-19. Tata McGraw Hill, New Delhi. pp. 134-38, 2004.

[19] Vijayanand Kommaluri, Sirajul Islam Choudhury, Pranab Ratna, "VAASAANUBAADA-Automatic Machine Translation of Bilingual Bengali-Assamese News Texts", *Langu-age Engineering Conference. Hyderabad, India.* 2002. [Online] Available: http://www.portal.acm.org/citation.cfm?id=788716.

[20] R. Moona Bharati, P. Reddy, B. Sankar, D.M. Sharma, R. Sangal,"Machine Translation: The Shakti Approach. Pre-Conference Tutorial", *ICON-2003*. [Online] Available: http://www.ebmt.serc.iisc.ernet.in/mt/login.html, [Online] Available: http://www.gdit.iiit.net/~mt/shakti.

[21] Sivaji Bandyopadhyay,"Use of Machine Translation in India", *AAMT Journal*, 36. pp. 25-31, 2004.

[22] S. Bandyopadhyay,"ANUBAAD - The Translator from English to Indian Languages", *In proceedings of the VIIth State Science and Technology Congress*. Calcutta. India. pp. 43-51, 2004.

[23] Saha Gautam Kumar, “The EB-ANUBAD translator- A Hybride Scheme", *Journal of Zhejjang University Science*, pp. 1047-1050, 2005.

[24] D. Gupta, N. Chatterjee,"Identification of Divergence for English to Hindi EBMT", *In proceedings of MT SUMMIT IX*. New Orleans, Louisiana, USA. pp. 157-162, 2003.

[25] Raghavendra Udupa, Tanveer A. Faruquie,"An English-Hindi Statistical Machine Translation System", *In proceedings of First International Joint Conference*, Hainan Island, China, March 22-24, pp. 254-262, 2004.

[26] B. K.Murthy, W. R.Deshpande,"Language technology in India: past, present and future", *In proceedings of MLIT Symposium 3. GII/GIS for Equal Language Opportunity*. Vietnam. October 6-7. Pp.134-137, 1998.

[27] Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing with Python, 1st ed., O’Reilly Media, June 2009.

Authors’ Profiles

Ruchika A. Sinhal was born in Wardha, India in 1988. She received B. E. in Computer Engineering from Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur and M. Tech in Computer Science and Engineering from RCOEM**,** Nagpur in 2010 and 2012 respectively. Her major field of study is natural language processing.

She is currently working as Assistant Professor in Shri Ramdeobaba College of Engineering and Management, Nagpur from 2013. She worked as Assistant Professor in Datta Meghe Institute of Engineering, Technology and Research, Sawangi(M), Wardha, from 2011. She has total 3 years of teaching experience. There are around 10 paper publication in my field.

Kapil O. Gupta was born in Karanja, India, in 1988. He received B. E. in Computer Engineering from Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur and M. Tech in Computer Science and Engineering from Sant Gadge Baba Amravati University, Amravati in 2010 and 2012 respectively.

He is currently working as assistant professor in Datta Meghe Institute of Engineering, Technology and Research, Sawangi(M), Wardha. He has total 3 years of teaching experience.

Mr. Gupta is member of professional societies like ISTE and IAENG. He has 4 papers in the areas of his interest.