Example Based Machine Translation Using Fuzzy Logic from English to Hindi

Manish Rana
Research Scholar, Post Graduate Department of Computer Science & Engineering,
Sant Gadge Baba Amravati University,
Amravati, India
manishrana@live.com

Mohammad Atique
Associate Professor,
Post Graduate Department of Computer Science,
Sant Gadge Baba Amravati University,
Amravati, India
mohd.atique@gmail.com

Abstract: *Example based machine translation is one of the approaches in machine translation. The concept uses the corpus of two languages and then translates the input text to desired target text by proper matching. The different languages have different language structure of the subject-object-verb (SOV) alignment. The matching is then arranged to give proper meaning in target text language and to form proper structure. This paper, describes the Example Based Machine Translation using Natural Language Processing demo. The proposed EBMT framework can be used for automatic translation of text by reusing the examples of previous translations through the use of Fuzzy which is proposed work. This framework comprises of three phases, matching, alignment and recombination. Same type of machine translation is possible through use of soft computing tool (Fuzzy Logic).*

Keyword Terms: Machine Translation; Machine Learning; Natural Language Processing; Fuzzy Logic; Fuzzification; Fuzzy Rules; Visualization; Visualize Data; subsuming; etc.

I. INTRODUCTION

Machine Learning or Machine translation is a key to future for Artificial Intelligence world. It is the first step toward the growth of AI machines. These machines will be communicating to use in our language (Natural Language Processing). The question is "Is it possible?" The answer is "yes", but "when", "How much Time "required making these type of machine.

Machine learning techniques are often used for financial analysis and decision-making tasks such as accurate forecasting, classification of risk, estimating probabilities of default, and data mining. However, implementing and comparing different machine learning techniques to choose the best approach can be challenging. Machine learning is synonymous with **Non-parametric** modeling techniques. The term non-parametric is not meant to imply that such models completely lack parameters but that the number and nature of the parameters are flexible and determined from data.
In this example, several supervised machine learning techniques available in MATLAB are highlighted. One may apply one or more of the techniques and compare them to determine the most suitable ones for different data sets.

II. RELATED WORK

Author describes here the translation of written texts or documents, not the interpretation of spoken utterances. The transfer of spoken to written language [7], and the synthesis of spoken language from written texts are topics in their own right which can be treated separately.

Machine translation of Indian Languages has been pursued mostly on the linguistic side. Hand crafted rules were mainly used for translation, [8][9]. Rule based approaches were combined with EBMT system to build hybrid systems and perform Interlingua based machine translation [10][11]. Input in the source language is converted into UNL, the Universal Networking Language and then converted back from UNL to the target language.

Recently, used linguistic rules are used for ordering the output from a generalized example based machine translation [12].While, in general in the machine translation literature, hybrid approaches have been proposed for EBMT primarily using statistical information most of which have shown improvement in performance over the pure EBMT system [13].

Automatically derived a hierarchical TM from a parallel corpus, comprising a set of transducers encoding a simple grammar [14]. Used example-based re-scoring method to validate SMT translation candidates and proposed an example based decoding for statistical machine translation which outperformed the beam search based decoder Showed improvement in alignment in EBMT using statistical dictionaries and calculating alignment scores bi-directionally combined the sub- sentential alignments obtained from the EBMT systems with word and phrase alignments from SMT to make 'Example based Statistical Machine Translation' and 'Statistical Example based Machine Translation' [14].

III. Machine Transaltion Process

1) Segmenting documents into words, sentences and formatting information

The basic elements of translation programs are words and rules for combining them to form sentences, paragraphs and complete texts. Every document to be translated first needs to be decomposed into words, numbers and punctuation marks. Since the layout of the translation in most cases should look just like the original, this information must also be recognized so it can be inserted into the translation at the proper places.

2) Reduction of word forms to their canonical form and dictionary lookup

Every translation program needs a dictionary. Here all information is stored which is necessary for the analysis of sentences and their translation, e.g. part of speech, gender, or semantic classification.

3) Recognizing sentential structures

In the beginning many researchers believed that could obtain reasonable translations by having a program translate word by word. It became clear very quickly that this was an illusion, because firstly, languages differ very much in word order, and secondly, many words can have more than one meaning of which only one is valid in a given sentence. The results were completely unintelligible sequences of alternate word translations which nobody could use.

4. Assigning translations to single words

Each word and many word groups are associated with one or more translations in the dictionary. When after grammatical analysis the contexts of the words are known, the appropriate translations can be selected.

5. Generating the structure of target sentences

Starting from the structure of the source sentence and the word translations selected, the structure of the target sentence is built up. It can be quite different from the original. Thus

e.g. Sentence : "*India has won the match by six wickets*"
Tokens : "*India" "has" "won" "the" "match" "by" "six" "wickets"*

Becomes

भारत छह विकेट से मैच जीत लिया है

Bhārata chaha vikēṭa sē maica jīta liyā ha

because the word vikēṭa in Hindi is not transitive, and therefore an additional verb - jīta - is required as a kind of intermediary.

6. Generating word forms

During the generation of the correct word order for the target sentence, translation programs usually work with canonical forms or word stems. Only after the structure established, forms such as vikēṭa and maica of the previous example become Bhārata chaha vikēṭa sē maica jīta liyā ha "भारत छह विकेट से मैच जीत लिया है".

7. Adding layout information

The layout information which was taken out in the first step must now be added to the translations such that in the end there is a new text which almost looks like the original. One note may be in order here: some formatting information such as bold face must be available even during the translation process, since the corresponding translations should appear in bold as well.

Machine translation, sometimes referred to by the abbreviation MT (not to be confused with translation, machine-aided human translation (MAHT) or interactive translation is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one natural language to another.

A. Tokenization

Tokenization is a primary step of Example based machine translation. In this phase, the input sentence is decomposed into tokens. These tokens are give n to POS stagger function to tag the tokens with their respective type.
e.g. Sentence : "*India has won the match by six wickets*"
Tokens : *"India" "has" "won" "the" "match" "by" "six" "wickets"*
भारत छह विकेट से मैच जीत लिया है

B. POS Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.
Matching Phase

Searching the source side of the parallel corpus for 'close' matches and their translations.. In matching phase, each token which is tagged with POS tag is searched in the dictionary of words and if match is found, then that word is passed to next phase**.**

C. Word-based Matching

Perhaps the "classical" similarity measure, suggested by Nagao (1984) and used in many early EBMT systems, is the use of a thesaurus or similar means of identifying word similarity on the basis of meaning or usage. Here, matches are permitted when words in the input string are replaced by near synonyms (as measured by relative distance in a hierarchically structured vocabulary, or by collocation scores such as mutual information) in the example sentences. This measure is particularly effective in choosing between competing examples, as in Nagao's examples [6], where, given (14a, b) as models, we choose the correct translation of *eat* in (15a) as *taberu* 'eat (food)', in (15b) as *okasu* 'erode', on the basis of

the relative distance from *he* to *man* and *acid*, and from *potatoes* to *vegetables* and *metal*.

D. Indexing

In order to facilitate the search for sentence substrings, we need to create an inverted index into the source-language corpus. To do this we loop through all the words of the corpus, adding the current location (as defined by sentence index in corpus and word index in sentence) into a hash table keyed by the appropriate word. In order to save time in future runs we save this to an index file.

E. Chunk searching and subsuming

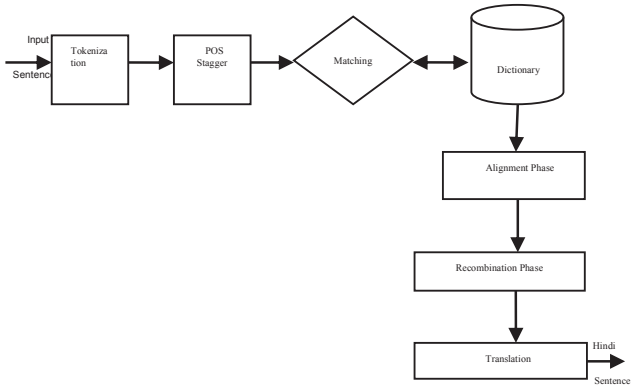


Figure: 1 Design of Proposed System for EMBT

A. Alignment Phase

Determining the sub sentential translation links in those retrieved examples.
The alignment algorithm proceeds as follows:
Stem the words of specified [1] source sentence look up those words in a translation dictionary Stem the words of the specified target sentence try to match the target words with the source words—wherever they match, mark the correspondence table. Prune the table to remove unlikely word correspondences. Take only as much target text as is necessary in order to cover all the remaining (unpruned) correspondences for the source language chunk. The pruning algorithm relies on the fact that *single* words are not often violently displaced from their original position

B. Recombination Phase

Recombining relevant [2] parts of the target translation links to derive the translation. Having matched and retrieved a set of examples, with associated translations, the next step is to extract from the translations the appropriate fragments ("alignment" or "adaptation"), and to combine these so as to produce a grammatical target output ("recombination"). This is arguably the most difficult step in the EBMT process [3]. difficulty can be gauged by imagining a source-language monolingual trying to use a TM system to compose a target

Keep two lists of chunks: current and completed.
Looping through all words in the target sentence:

See whether locations for the current word extend any chunks on the current list. If they do, extend the chunk.
Throw away any chunks that are 1-word. These are rejected.
Move to the completed list those chunks that were unable to continue. Start a new current chunk for each location
At the end, dump everything into completed.
Then, to prune, run every chunk against every other:
If a chunk properly subsumes another, remove the smaller one
If two chunks are equal and we have too many of them, remove one.

IV. Proposed Approach

A. Translation

In this phase, after matching and recombination, the matched words are mapped with the Hindi Corpus by searching. If it finds match, the Hindi word is substituted.
EBMT Implementation

Example Based Machine Translation is based on the idea to reuse the previously done translations as examples. Following are three examples are given. EBMT system tries to translate the given input English Text into Hindi by using these previous translations.

B. Graphical User Interface (GUI)

The Graphical User Interface is prepared for the project Example Based Machine Translation by using MATLAB GUI Editor utility.

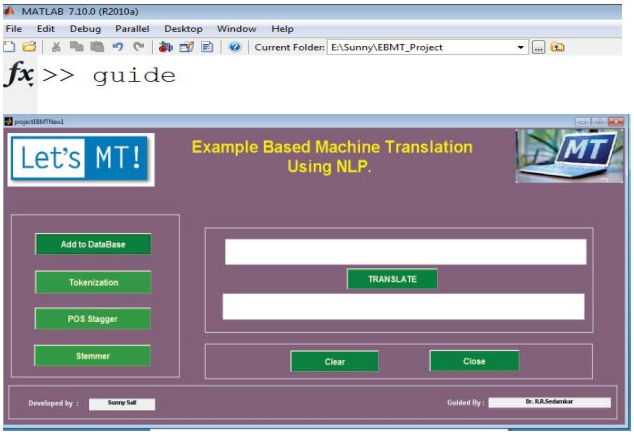


Figure2. Screen shot for User Interface

1. Add to Database

This option is used to add new word [5] and its corresponding Hindi word into corpus or dictionary. New English word along with its type i.e Noun, Pronoun, Verb, Adjective and Adverb can be added by using this form.

Figure 3. Add to dictionary

i	मै	noun
we	हम	noun
you	तुम	noun
you	तुम	noun
he	वो	pronoun
she	वो	pronoun
it	वह	pronoun
am	हू	verb
is	है	verb
are	है	verb
was	था	verb
were	थे	verb
have	है	verb
has	है	verb
has	है	verb
had	थी	verb
table	मेज	noun
cloths	कपडे	noun
india	भारत	noun
river	नदी	noun
water	पानी	noun
country	देश	noun
big	बडा	adjective
good	अच्छा	adjective
bad	बुरा	adjective

fresh	ताजा	adverb
this	ये	pronoun
that	वो	pronoun
door	दरवाजा	noun
medicine	दवा	noun
like	पसंद	verb
cricket	क्रिकेट	noun
small	छोटा	adjective
small	छोटा	adjective
broad	चौडा	adjective
game	खेल	noun
house	घर	noun
home	घर	noun
go	जाना	verb
come	आना	verb
walk	चलना	verb
walking	चल रहा	verb
very	बहुत	adjective
Sunny	धनी	noun
chating	वार्तालाप	verb
slowly	धीरे	adverb
great	महान	adjective
God	भगवान	noun
Ram	राम	noun
god	भगवान	noun

Figure : 4.Dictionary

2. Tokenization

Tokenization is a primary step of Example based machine translation. In this phase, the input sentence is decomposed into tokens. These tokens are give n to POS stagger function to tag the tokens with their respective type.

```
Input String :'India is my country. It is a great country!'

ring =

is my country. It is a great country!


 are 2 sentences in this paragraph !

 The Tokens are......
------------------------------------------------------------
```

```
token =

    'India'    ' is'    ' my'    ' country.'          []
    'It'       ' is'    ' a'     ' great'       ' country!'
```

Figure: 5 Result of tokenization

3. POS Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.

```
inpPString =

Boy is very good

-----------------------------------------------------

 boy :  noun

 is :   verb

 very :     adjective

 good :     adjective
-----------------------------------------------------
```

Figure: 6 POS Tagger result

4. Stemmer :

This option is very useful to find stem i.e root word of any word :
e.g. for “running” stem is **run**
for “Indian” stem is **India**,
for “beautiful” stem is **beauty**.

```
Indian useful beutiful stemmer

----------------------------------------------------

 Stem = india
stem =

use


 Stem = use
 Stem = beuty
 Stem = stem
```

Figure: 7. Result of Stemmer

5. Translation

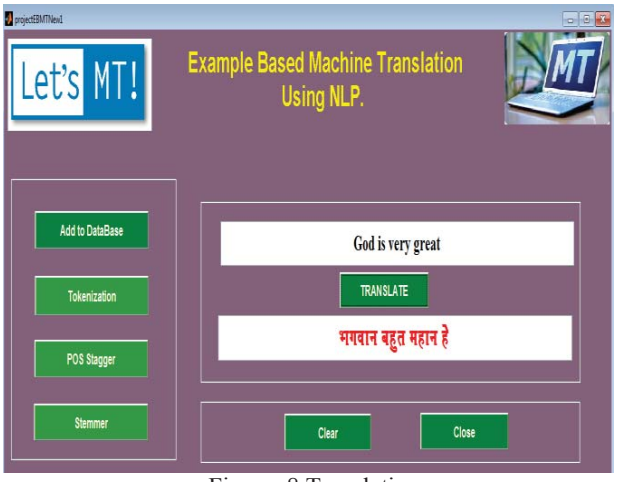


Figure: 8 Translations

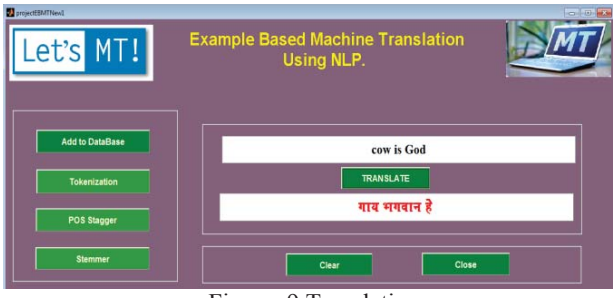
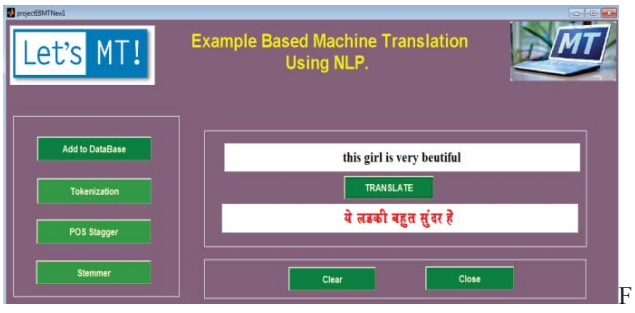


Figure: 9 Translations



F

Figure: 11 Translations

Example 1
English : India won the match.
Hindi : भारत ने मैच जीता

Example 2
English : India is the best
Hindi : भारत सबसे अच्छा है

Example 3
English : Sachin plays well
Hindi : सचिन अच्छी तरह से खेलते हैं

Input
English : Sachin is the best
Translation (Output)
Hindi : सचिन सबसे अच्छा है

Table 1 illustrates the comparison of three machine translation techniques, Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT) and Example-Based Machine Translation (EBMT) on the basis of various parameters such as Consistency, predictable quality, Quality of out of domain translation, Use of grammar, robustness, Fluency and performance.

Table: 1 Comparison of various Machine Translation schemes

Parameter	**RBMT**	**SMT**	**EBMT**
Consistency	High	Low	Medium
Predictable Quality	Good	Similar	Very well
Out of Domain Quality	Medium	Low	High
Use of Grammar	Yes	No	No
Robust	Yes	No	Yes
Fluency	Less	Medium	High
Performance	Good	Medium	Good

Some translation results are also presented of some existing machine translation tools and there system in Table 2.

Table: 2 Comparison of translation of text from various Machine Translation tools

English Sentence	**Hindi Translation by existing MT tools**	**Hindi translation by our EBMT translation**
India is great	भारत है महान	भारत महान है
I am a boy	मैं हूँ लड़का	मैं लड़का हूँ
She is beautiful	वह है सुंदर	वह सुंदर है
He was great	वह है महान	वह महान है
Where do you live ?	आप हैं कहाँ रहते?	आप कहाँ रहते हैं?
She reads book	वह किताब है पढ़ता	वह किताब पढ़ता है
Milk is white	दूध है सफेद	दूध सफेद है

V. RESULTS

Based on this model expected result is as following:
Input: Source Language, **ENGLISH** (SL).
Output: Target Language, **HINDI** (TL).
The result obtained is with minimal human interface.

Table: 3 Performance evaluation of EBMT

Corpus Size (No. of Sentences)	Unigram Precision	Unigram Recall	F-measure	BLEU	NIST	mWER	SSER
50	0.71	0.79	0.74	0.71	2.6	81.11	94.21
100	0.74	0.80	0.76	0.73	3.2	78.44	93.96
200	0.79	0.85	0.81	0.75	3.9	77.24	93.12
300	0.84	0.88	0.85	0.81	4.5	74.02	92.32
500	0.85	0.92	0.88	0.83	5.0	70.00	89.44
1000	0.90	0.94	0.91	0.91	6.6	65.22	81.77

VI. CONCLUSION & FUTURE SCOPE

This research work proposes a new system, which is scalable, transparent and efficient. The entire system will convert the source language text into target language text using natural language processing. It will use the machine translation technique which is better than the existing tools available in the market.

The algorithm is such that, there is dictionary / corpus / vocabulary of **English** and **Hindi**. The parsing will be proper. The mapping technique will also be used. All the Literals will be separated using partitioning and stemming techniques. The root word will be identified using artificial intelligence and bilingual translation.

We pursue the study of example based machine translation using natural language processing.

VII. REFERENCE

[1] Ali, V. Singh, "Potentials of Fuzzy Logic : An Approach To Handle Imprecise Data," *American Medical Informatics Association*, vol 2, no. 4, pp. 358- 361, 2010.

[2] E. Binaghi, I. Gallo, C. Ghiselli, L. Levrini, K. Biondi, "An Integrated Fuzzy Logic And Web-Based Framework For Active Protocol Support," *International Journal of Medical Informatics*, vol. 77, pp. 256-271, 2008.

[3] A. Ciabattoni, T. Vetterlein, K. K. Adlassnig, "A Formal Logical Framework for Cadiag-2," *Studies in Health Technology & Informatics*, vol. 150, pp. 648- 652, 2009.

[4] H. Owaied, M. M. Qasem, "Developing Rule-Case- Based Shell Expert System," *Proc.of Int. MultiConf. of Engineers & Scientists*, vol. 1,pp.234-240, 2010.

[5] M. G. Tsipouras, C. Voglis, D. I. Fotiadis, " A Framework for Fuzzy Expert System Creation -Application to Cardiovascular Diseases," *IEEE Transactions Biomedical Engg*., vol. 54, no. 11, pp. 2089-2105, 2007.

[6] X. He and L. Deng, "Speech-centric information processing: An optimization-oriented approach," Proc. IEEE, vol. 101, no. 5, pp. 1116–1135, May 2013.Zadeh LA: Fuzzy sets. Information and control 8: 338-353, 1965 A system architecture for medical informatics. Fuzzy sets and systems 1994 ; 66 : 195-205.

[7] C. Liu, Y. Hu, L.-R. Dai, and H. Jiang, "Trust region-based optimization for maximum mutual information estimation of hmms in speech recognition," IEEE Trans.Audio,Speech,Lang. Process., vol. 19, no. 8, pp. 2474–2485, Nov. 2011.

[8] F. Castro, A. Vellido, A. Nebot, and F. Mugica, ''Applying data mining techniquestoe-learningproblems,''inEvolutionofTeachingandLearning Paradigms in Intelligent Environment. Berlin, Germany: Springer-Verlag, 2007, pp. 183–221.

[9] Hongshen Chen, Jun Xie, Fandong Meng, Wenbin Jiang Qun Liu "*A Dependency Edge- based Transfer Model for Statistical Machine Translation*" Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, August 23-29 2014, pages 1103–1113

[10] J.Baker, L. Deng,J.Glass,S. Khudanpur,C.-H.Lee,N.Morgan,and D. O'Shgughnessy, "Research developments and directions in speech recognitionandunderstanding,parti,"IEEESignalProcess.Mag.,vol. 26, no. 3, pp. 75–80, 2009.

[11] H.Zen,M.J.F.Gales,Y.Nankaku,andK.Tokuda,"Productofexperts forstatisticalparametricspeechsynthesis,"IEEEAudio,Speech,Lang. Process., vol. 20, no. 3, pp. 794–805, Mar. 2012.

[12] L. Deng, "Switching dynamic system models for speech articulation and acoustics," in Mathematical Foundations of Speech and LanguageProcessing. New York, NY, USA: Springer-Verlag, 2003, pp. 115–134

[13] G.Dahl,D.Yu,L.Deng,andA.Acero,"Context-dependentpre-trained deep neural networks for large-vocabulary speech recognition,"IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 1, pp. 30–42, Jan. 2012.

[14] D.GolovinandA.Krause,"Adaptivesubmodularity:Anewapproach to active learning and stochastic optimization," in Proc. Int. Conf. Learn. Theory, 2010.

Author Detail:

Manish Rana, PhD scholar Department of Computer Science, Sant Gadge Baba Amravati University, Amravati. He has more than seven years Teaching Experience. His area of interest includes Artificial Intelligence, Machine translation and soft computing. He has published 5 International Journal and 3 Papers in national Conference.

Dr. Mohammad Atique , is presently working as Associate Professor, P.G. Department of Computer Science, Sant Gadge Baba Amravati University, Amravati. He has around 37 publications to his credit in International/National Journal and conferences. His area of interest includes Artificial Intelligence, Machine translation and soft computing.