

**Computer Aided Translation
&
Automatic Post Editing**

A thesis

Submitted in partial fulfillment of the requirement for the Degree of
Master of Computer Science and Engineering
of
Jadavpur University

By

Tapas Nayak

Registration No.: 92922 of 2005-6

Examination Roll No.: M4CSE1610

Under the Guidance of

Dr. Sudip Kumar Naskar

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2016

**Computer Aided Translation
&
Automatic Post Editing**

A thesis

Submitted in partial fulfillment of the requirement for the Degree of
Master of Computer Science and Engineering
of
Jadavpur University

By

Tapas Nayak

Registration No.: 92922 of 2005-6

Examination Roll No.: M4CSE1610

Under the Guidance of

Dr. Sudip Kumar Naskar

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2016

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that the dissertation entitled “Computer Aided Translation & Automatic Post Editing” has been carried out by Tapas Nayak (University Registration No.: 92922 of 2005-6, Examination Roll No.: M4CSE1610) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Master of Computer Science and Engineering. The research results presented in the thesis have not been included in any other thesis submitted for the award of any degree in any other University or Institute.

.....
Dr. Sudip Kumar Naskar(Thesis Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

Countersigned

.....
Prof. Debesh Kumar Das
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-32.

.....
Prof. Sivaji Bandyopadhyay
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-32.

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval*

This is to certify that the thesis entitled “Computer Aided Translation & Automatic Post Editing” is a bona-fide record of work carried out by Tapas Nayak in partial fulfillment of the requirements for the award of the degree of Master of Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2015 to May 2016. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

.....
Signature of Examiner 1

Date:

.....
Signature of Examiner 2

Date:

*Only in case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled “Computer Aided Translation & Automatic Post Editing” contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Science & Engineering.

All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Tapas Nayak

Registration No: 92922 of 2005-6

Exam Roll No.: M4CSE1610

Thesis Title: Computer Aided Translation & Automatic Post Editing

.....
Signature with Date

Acknowledgement

I would like to start by thanking the holy trinity for helping me deploy all the right resources and for shaping me into a better human being.

I would like to express my deepest gratitude to my advisor, **Dr. Sudip Kumar Naskar**, Assistant Professor, Department of Computer Science and Engineering, Jadavpur University for his admirable guidance, care, patience and for providing me with an excellent atmosphere for doing research. Our numerous scientific discussions and his many constructive comments have greatly improved this work.

I would like thank **Santanu Pal**, PhD fellow at Saarland University for his guidance throughout my master's program and providing me valuable research ideas for computer aided translation word and automatic post editing work.

Words cannot express my indebtedness to **Prof. Sivaji Bandyopadhyay**, Department of Computer Science and Engineering, Jadavpur University who is also Dean, Faculty of Engineering and Technology, Jadavpur University for his amazing guidance and supervision. I am deeply grateful to him for the long discussions that helped to enrich the technical content of this manuscript. Without his enthusiasm, encouragement, support and continuous optimism this thesis would hardly have been completed.

I would like to thank **Prof. Debesh Kumar Das**, Head, Department of Computer Science and Engineering, Jadavpur University for providing me with moral support at times of need.

Most importantly none of this would have been possible without the love and support of my family. I extend my thanks to my parents, especially to my mother whose forbearance and whole hearted support helped this endeavor succeed.

This thesis would not have been completed without the inspiration and support of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

.....
Tapas Nayak
Registration No: 92922 of 2005-6
Exam Roll No.: M4CSE1610
Department of Computer Science & Engineering
Jadavpur University

Contents

Chapter 1	Introduction.....	12
1.1	Statistical Machine Translation.....	13
1.2	Translation Memory.....	14
1.3	Computer Aided Translation.....	14
1.4	Automatic Post Editing.....	15
1.5	Motivation.....	15
1.6	Thesis Organization.....	16
Chapter 2	Statistical Machine Translation.....	17
2.1	Theory of Statistical Machine Translation.....	17
2.1.1	Language Model.....	18
2.1.2	Translation Model.....	20
2.2	Machine Translation Tools.....	23
2.2.1	Language Model Tools.....	24
2.2.2	Translation Model Tools.....	24
2.2.3	Translation Engine.....	25
2.3	MT System Evaluation.....	25
2.3.1	BLEU.....	26
2.3.2	WER.....	27
2.3.3	Meteor.....	28
2.3.4	TER.....	28
Chapter 3	Computer Aided Translation.....	29
3.1	CAT Tools.....	29
3.1.1	SDL Trados.....	29
3.1.2	Wordfast.....	30
3.1.3	PET.....	31
3.1.4	MateCat.....	31
3.1.5	CasMaCat.....	31
3.2	Limitations of Current CAT Tools.....	32

Chapter 4	CATaLog - A TM based CAT Tool.....	33
4.1	Related Research Work.....	34
4.2	System Description	36
4.3	Corpus	37
4.4	Similarity Metric	38
4.5	Color Coding.....	39
4.6	Improving TM Search Time	40
4.6.1	Inverted Index.....	40
4.6.2	Length Based Pruning.....	42
4.7	Bulk Translation Facility	42
4.8	Controlling Application Parameters.....	43
Chapter 5	Mismatched Segments (MS) Fusion in CATaLog	45
5.1	Related Work	46
5.2	System Description	48
5.3	Generating Dictionary.....	48
5.4	Grouping Parts Of Speech Tags.....	51
5.5	Finding Translations for Unmatched Parts	51
5.6	Finding Positions to Insert Translations.....	53
5.7	Matching using POS N-grams	53
5.8	Parse Tree Matching	55
5.9	Illustration with an Example	56
5.10	Length Based Pruning	61
5.11	Re-ranking of TM Suggestions	62
5.12	Experiments and Results	64
Chapter 6	Automatic Post Editing	69
6.1	Related Research Work.....	69
6.2	Dataset.....	72
6.3	Baseline APE System.....	72
6.4	Common Errors in SMT Output	73
6.5	Our Approach.....	75

6.5.1	Word Alignment Based MT Word Deletion Model	75
6.5.2	Source Word Context Based MT Word Deletion Model	78
6.5.3	Word Deletion Model	80
6.5.4	Surface Form Correction Model	80
6.5.5	Combination of Word Deletion Model and Surface Form Correction Model	82
6.5.6	Automatic Post Editing Model	82
Chapter 7	Conclusions.....	84
7.1	Scope of future work.....	84
Bibliography	86

List of Figures

Figure 1 : CATaLog Main Screen	37
Figure 2 : Bulk Translation Form	43
Figure 3 : Parameter Settings Form	44
Figure 4 : Parse Tree.....	59

List of Tables

Table 1 : Test Sentence and TM Suggestion Alignment	57
Table 2 : Result of CATaLog System.....	67
Table 3 : APE Baseline System Result.....	73
Table 4 : MT Word Deletion Statistics Format	75
Table 5 : Result of APE Deletion Experiment.....	77
Table 6 : Result of APE Deletion Experiment with Word Stemming.....	78
Table 7 : Result of Source Word Context based MT Word Deletion Model	79
Table 8 : Result of Combination of Word Deletion Model	80
Table 9 : Result of Surface Form Correction Model	81
Table 10 : Result of Word Deletion and Surface Form Correction Model	82
Table 11 : Result of APE Model.....	83

Introduction

Machine translation research can be traced back to 1950 when US technological giant IBM tried to translate from Russian to English automatically. But in the mid 60s an adverse report from US government committee led to withdrawal of US government funding to machine translation research. Committee reported that human translation is much more accurate and cost effective than machine generated translation. Though lack of funding slowed the research growth in machine translation areas, but it was not stopped totally. During this time only a rule based machine translation system called 'SYSTRAN' was developed by researcher. This system was the starting point for most of the modern day translation system like Google Translate, Babel Fish etc.

Many professional machine translation systems are available in today's era. Quality of machine translation output has improved a lot since 1950s, but still it is not good enough to generate publishable content using MT system. Researcher came up with a little different idea to generate quality translation from an already human translated database called 'Translation Memory'. Translation memory stored millions of source sentences and their human translations as a database. New sentence are matched with translation memory database and its translation is shown to the user. Translation memory output quality is good, but many researchers don't consider it as machine translation system, since it is just string matching system, no proper translation logic is there.

Translation memory works well when sentences are repetitive. But it fails miserably when that is not the case. So translation industry goes back to use machine

translation system again, but they post edit the machine generated output by human translator to make it publishable. This task of post editing is very tedious and time consuming and very much costly. Researchers started work to build some tools which will help human translator in their work which are called 'Computer Aided Translation' tool. Many commercial and free CAT tools are available now which are used by human translator extensively for their work. These tools make their life quite easier.

Post editing of machine generated translation lead to research of doing that post editing task by machine itself. Many researchers are trying to emulate the human post editor behavior to automate the post editing task. This research area is called 'Automatic Post Editing'.

1.1 Statistical Machine Translation

Initially machine translation system was built based on translation rules. Rules were written in a file and then those rules were applied on source sentence to generate target sentence. Separate rule set was required for every language pair. This process is very simple, but translation produced by this type of system was very bad. Also maintaining rules was very difficult. One has to add new rules every day. So this method was abandoned by almost all MT research group.

Statistical machine translation has come up as alternative to rule based machine translation. Research in this area was started in 1950s, but significant improvement was made in 1990s by IBM. IBM proposed first statistical model based machine translation system which is called Model 1. After that many improvements have been made on that and IBM released Model 2 to Model 5. All these models are based on Bayes theorem.

1.2 Translation Memory

‘Translation Memory’ is a database of bi-lingual sentences. It stores millions of source language sentences and their human translation. Everyday new sentences and their translation are added to it to make it more robust. Translation memory can be used to in domain specific translation field where there is not many variations are seen in source sentences. If source sentences are repetitive or they are very closely similar to some old sentences, translation memory can give very good result. Computer Aided Translation systems use translation memory extensively.

Translation memory contains millions of sentence pair which is translated by humans. One can easily understand that building this kind of huge TM is very much costly. To make it robust, one has to update the TM database daily basis which makes it more expensive.

When a new sentence comes, it is matched against all the sentences available in translation memory. If a match found, its corresponding translation is shown as output. Since TM contains millions of sentence pair, these matching task highly expensive in terms of time. Software which uses translation memory needs to improve this searching time using various techniques like indexing, dictionary etc.

1.3 Computer Aided Translation

After decades of research in Machine Translation, researchers are unable to build a MT system whose output can be used in publishing. MT outputs can be useful for internal use or may be for those areas where rough translation will be good enough. But raw MT outputs can’t be published for larger public. That is the reason MT outputs needs to be post edited by professional post editors. Task of post editing is a very tedious and time consuming also. It is very costly too. Post editors charge for

every word needs to be post edited. Computer Aided Translation (CAT) tools are used to help post editors in their task. CAT tools first give some translation suggestions and also provides graphical helps to reduce post editing effort. It has two major components – Machine Translation (MT) System and Editing Interface. MT system generates number of suggestions for the sentence to be translated. Interactive GUI helps post editors to edit one of the many suggestions and generate proper translation.

1.4 Automatic Post Editing

Automatic Post Editing is new field in MT and CAT domain. Here rather than using some human post editors, automatic system tries to transform MT output to a new translation which will be more similar to post edited one. This type of system uses one source file, its MT output and Human Post Edited output for training. Sometimes we can use trace of the MT system for training purpose, if it is available. Most of the time, this trace is not available. One has to think the MT system as a black box. To use this system, Source sentence and its MT output is given as input and system generates the PE translation.

1.5 Motivation

Some CAT tools are already available which is used by human post editors to reduce their effort and save time. But these tools are mostly proprietary software and they cost a huge amount. Our aim is to develop one such CAT tool which will available for free, so that human post editors can use them for their job. We will try to provide many features on user interface perspective, which will make the post editing task more interesting.

Automatic Post Editing is a very new and challenging research area in machine translation field. In this researchers try to automate the human post editing work. If

we can emulate the human post editing task and generate the post edited output automatically, we can eliminate the entire human post editing effort from machine translation chain. It will save lot of time and money too. So I will put some effort on this aspect also.

1.6 Thesis Organization

The **Chapter 1** of the thesis contains the preliminary concepts of machine translation, statistical machine translation, translation memory, computer aided translation and automatic post editing. In **Chapter 2** explains the statistical machine translation in details. It explains how a SMT system works, what are the tools available to build a baseline SMT system, how to evaluate machine translation system. **Chapter 3** explains details about computer aided translation, different CAT tools available in market. **Chapter 4** explains about our translation memory based computer aided translation tool CATaLog. **Chapter 5** describes how we can improve TM suggestions by fusion of mismatched parts of input sentence using POS tags and Parse tree. **Chapter 6** briefly explains about automatic post editing and different ways we can emulate human post editing task. Finally **Chapter 7** contains the conclusion and future scope of this work.

Statistical Machine Translation

Rule based machine translation didn't give desired results in the field of translation. Major problem with RBMT system was for different language pair, researcher had to develop new system. Along with that, rules needs to be updated regularly which was quite time consuming and may be erroneous also. Research has shown that adding too many rules in RBMT system may not lead to better translation, but actually it may downgrade the system performance.

This is the reason why researchers have come up with statistical machine translation model which is language independent. Naïve Bayes theorem has been used to model this kind of MT system. It uses mainly three sub model- language model, translation model and distortion model to generate a possible translation which retains the meaning of source sentence and this translation is quite smooth in target language. Many research results have shown that translation generated by SMT system is much better than any other MT models.

2.1 Theory of Statistical Machine Translation

Lets F represent a foreign language sentence and E is its corresponding English translation. $P(E|F)$ is the probability of E being the translation of F. If there n possibilities of English translation, we will get n probabilities. Whichever E will give the highest probability that will be the best English translation of F.

$$\hat{E} = \underset{E \in \text{English}}{\operatorname{argmax}} P(E | F)$$

If we apply Bayes theorem the equation we get is as follows

$$\begin{aligned}
\hat{E} &= \operatorname{argmax}_{E \in \text{English}} P(E | F) \\
&= \operatorname{argmax}_{E \in \text{English}} \frac{P(F | E)P(E)}{P(F)} \\
&= \operatorname{argmax}_{E \in \text{English}} P(F | E)P(E)
\end{aligned}$$

Ignoring the denominator part since it is same for all possible English translation.

Here $P(E)$ is called the language model probability and $P(F|E)$ is called translation model probability.

2.1.1 Language Model

Language model $P(E)$ can be trained on large, unsupervised monolingual corpus of target language E . One can use N-gram model for training. Language model assigns a probability to a sentence in target language. It actually tells how smooth one sentence is in target language.

Let's assume $E = E_1 E_2 \dots E_M$

$$\begin{aligned}
P(E) &= P(E_1 E_2 \dots E_M) \\
&= \prod_{i=1}^M P(E_i | E_1 E_2 \dots E_{i-1})
\end{aligned}$$

$$P(E_i | E_1 E_2 \dots E_{i-1}) = \text{Count}(E_1 E_2 \dots E_i) / \text{Count}(E_1 E_2 \dots E_{i-1})$$

But we may not get too many sentences in our corpus which matches $E_1 E_2 \dots E_{i-1}$. We will use Markov assumption that current word E_i depends on previous $N-1$ words $E_{i-1} \dots E_{i-N+1}$. This is called N-gram model. We can choose N to be 1 or 2 or 3 which are called unigram or bigram or trigram model.

$$\text{Unigram} \quad P(E_i | E_1 E_2 \dots E_{i-1}) = P(E_i)$$

$$= \text{Count}(E_i) / \text{Count}(\text{token})$$

$$\begin{aligned} \text{Bigram} \quad P(E_i | E_1 E_2 \dots E_{i-1}) &= P(E_i | E_{i-1}) \\ &= \text{Count}(E_{i-1} E_i) / \text{Count}(E_{i-1}) \end{aligned}$$

$$\begin{aligned} \text{Trigram} \quad P(E_i | E_1 E_2 \dots E_{i-1}) &= P(E_i | E_{i-2} E_{i-1}) \\ &= \text{Count}(E_{i-2} E_{i-1} E_i) / \text{Count}(E_{i-2} E_{i-1}) \end{aligned}$$

While calculating all this N-gram probabilities, one has to handle with different issues. If one use trigram model, there may be cases that for a particular sentence many words may not have trigrams present in training corpus. In that case we can move to bigram from trigram, if bigram exists. If no bigram, then we can move to unigram model. We can use interpolation of different N gram models.

Another issue countered in language model is about unknown words. If a word in new sentence is not available in training corpus, then its probability will be zero. In that case probability of that sentence will be zero, which is not a correct estimation. We can use smoothing also to get rid of this problem. There are several smoothing algorithm available like Add-1 smoothing, Add-k smoothing, Good-Turing smoothing, Kneser Ney smoothing, Witten-Bell smoothing. In smoothing technique we try to estimate the probability of unseen words based on those words which we have seen just once. So basically we are distributing the probability of words seen just once to the unseen words. Since total probability is 1, we need to reduce the probability of words seen once, twice, thrice so on.

2.1.2 Translation Model

For translation model one needs parallel corpora of target language and source language. These corpora will contain the source language sentences and their corresponding translation. But we don't know which target words or phrases are translation of which source words since these corpora is totally untagged. One intuition is that if a phrase combination of target phrase and source phrase appear in more than one sentence then we can assume that they are translation of each other. Using this intuition if we use expectation maximization algorithm, we can get the translation alignment between target language and source language. Phrase based alignment gives good result for natural languages. We can use same algorithm for word based alignment too.

Phrase based translation works in three steps.

$P(F | E)$ is modeled by translating phrases in E to phrases in F .

1. First segment E into a sequence of phrases $\bar{e}_1, \bar{e}_2, \dots, \bar{e}_I$
2. Then translate each phrase \bar{e}_i , into f_i , based on **translation probability** $\phi(f_i | \bar{e}_i)$
3. Then reorder translated phrases based on **distortion probability** $d(i)$ for the i th phrase. (distortion = how far the phrase moved)

$$P(F | E) = \prod_{i=1}^I \phi(\bar{f}_i, \bar{e}_i) d(i)$$

Assuming a **phrase aligned** parallel corpus is available or constructed that shows matching between phrases in E and F . Then compute (MLE) estimate of ϕ based on simple frequency counts.

$$\phi(\bar{f}, \bar{e}) = \frac{\text{count}(\bar{f}, \bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f}, \bar{e})}$$

Distortion probability is a measure of distance between positions of a corresponding phrase in the 2 languages. “What is the probability that a phrase in position X in the English sentences moves to position Y in the Spanish sentence?” Measure distortion of phrase i as the distance between the start of the f phrase generated by $\bar{e}_i, (a_i)$ and the end of the end of the f phrase generated by the previous phrase $\bar{e}_{i-1}, (b_{i-1})$. Typically assume the probability of a distortion decreases exponentially with the distance of the movement.

$$d(i) = c\alpha^{|a_i - b_{i-1}|}$$

Set $0 < \alpha < 1$ based on fit to phrase-aligned training data. Then set c to normalize $d(i)$ so it sums to 1.

Directly constructing phrase alignments is difficult, so rely on first constructing word alignments. We can learn to align from supervised word alignments, but human-aligned bi-texts are rare and expensive to construct. Typically use an unsupervised EM-based approach to compute a word alignment from un-annotated parallel corpus. To simplify the problem, typically assume each word in F aligns to 1 word in E (but assume each word in E may generate more than one word in F). Some words in F may be generated by the NULL element of E . Therefore, alignment can be specified by a vector A giving, for each word in F , the index of the word in E which generated it.

IBM Model 1 is the First model proposed in seminal paper by Brown *et al.* in 1993 as part of CANDIDE, the first complete SMT system.

- Assumes following simple generative model of producing F from $E = e_1, e_2, \dots, e_l$
- Choose length, J , of F sentence: $F = f_1, f_2, \dots, f_j$

- Choose a 1 to many alignment $A = a_1, a_2, \dots, a_J$
- For each position in F , generate a word f_j from the aligned word in E :
 e_{a_j}

Assume some length distribution $P(J | E)$ and all alignments are equally likely. Since there are $(I + 1)^J$ possible alignments:

$$P(A | E) = P(A | E, J)P(J | E) = \frac{P(J | E)}{(I + 1)^J}$$

Assume $t(f_x, e_y)$ is the probability of translating e_y as f_x , therefore:

$$P(F | E, A) = \prod_{j=1}^J t(f_j, e_{a_j})$$

Determine $P(F | E)$ by summing over all alignments:

$$P(F | E) = \sum_A P(F | E, A)P(A | E) = \sum_A \frac{P(J | E)}{(I + 1)^J} \prod_{j=1}^J t(f_j, e_{a_j})$$

Goal is to find the most probable alignment given a parameterized model.

$$\begin{aligned} \hat{A} &= \operatorname{argmax}_A P(F, A | E) \\ &= \operatorname{argmax}_A \frac{P(J | E)}{(I + 1)^J} \prod_{j=1}^J t(f_j, e_{a_j}) \\ &= \operatorname{argmax}_A \prod_{j=1}^J t(f_j, e_{a_j}) \end{aligned}$$

Since translation choice for each position j is independent, the product is maximized by maximizing each term:

$$a_j = \underset{0 \leq i \leq I}{\operatorname{argmax}} t(f_j, e_i) \quad 1 \leq j \leq J$$

IBM model 1 can be trained on a parallel corpus to set the required parameters. For supervised (hand-aligned) training data, parameters can be estimated directly using frequency counts. For unsupervised training data, EM can be used to estimate parameters, e.g. Baum-Welch for the HMM model.

Randomly set model parameters. Make sure they represent legal distributions

Until converge (i.e. parameters no longer change) do:

E Step: Compute the probability of all possible alignments of the training data using the current model.

M Step: Use these alignment probability estimates to re-estimate values for all of the parameters.

Phrase-based approaches to MT have been shown to be better than word-based models. However, alignment algorithms produce one to many word translations rather than many to many phrase translations. Combine $E \rightarrow F$ and $F \rightarrow E$ word alignments to produce a phrase alignment.

2.2 Machine Translation Tools

There are many open source SMT system available which can be used by researcher to build a baseline system. This system includes many language models, translation models and distortion models. Researches can choose proper model based on their requirement.

2.2.1 Language Model Tools

Language model toolkits are used to build N-Gram monolingual language models. There are many popular LM toolkits available to use in research work. SRILM (Stolcke, 2002) is a popular language model toolkit to make N-gram language models. It uses trie data structure to build language models. It stores N-gram probabilities in logarithmic form, so all probabilities are negative.

IRSTLM (Federico et al., 2008), KenLM (Heafield, 2011), RandLM (Talbot and Osborne, 2007), BerkeleyLM (Pauls and Klein, 2011) are some more popular language models used for research purpose. Language models are used to determine the smoothness of the translation in target language. To achieve good smoothness language models have to be very large. Efficient data structure like trie, sorted trie have to be used to build language models on large monolingual corpus.

2.2.2 Translation Model Tools

Translation models are used to generate the source word to target word alignment between bilingual corpus. There are many tools available to generate this alignment. *GIZA++*¹ is one such tool which is used widely. This is an extension of the program GIZA (part of the SMT toolkit EGYPT²) which was developed by the Statistical Machine Translation team during the summer workshop in 1999 at the Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). Berkeley Aligner³ is another tool which is also very popular in statistical machine translation field to generate unsupervised word alignment.

¹ <http://www.statmt.org/moses/giza/GIZA++.html>

²

<http://web.archive.org/web/20100215160706/http://www.clsp.jhu.edu/ws99/projects/mt/toolkit/>

³ <http://nlp.cs.berkeley.edu/software.shtml>

2.2.3 Translation Engine

Moses⁴ is a free statistical machine translation engine that can be used to train statistical models of text translation from a source language to a target language. Moses includes many language models like SRILM, IRSTLM, RandLM, KenLM etc. to build monolingual language models. It includes GIZA++, Berkeley aligner to generate translation model. Researcher can plug their own language model or translation model into Moses also. In that way Moses gives the researcher lot of flexibility. It uses beam search method to decode sentences after training. Beam search reduces the decoding search space significantly and produces good translation in quick time.

2.3 MT System Evaluation

Evaluation of machine translation output by human is the best way of evaluation, but is time-consuming and expensive. Automated evaluation comparing the output to multiple human reference translations is cheaper and correlates with human judgments.

Humans can be asked to evaluate MT output on several dimensions:

- **Fluency:** Is the result grammatically correct? Is it understandable, and readable in the target language?
- **Fidelity:** Does the result correctly convey the information in the original source language?

Evaluation of machine translation output can be done using computer aided translation tool. CAT tools can give measures like how many edit operations must be done on MT output to produce the correct translation. Edit operations can be measured either on word level or character level. It includes insertion of a word or

⁴ <http://www.statmt.org/moses/>

character, deletion of word or character or substitution of word or character. CAT tools can also take into account the time taken to convert the MT output to correct translation, number of keystrokes needed to achieve that.

Automatic Evaluation of MT output can be done in following ways:

- Collect one or more human *reference translations* of the source.
- Compare MT output to these reference translations.
- Give a score based on similarity to the reference translations.
- If multiple MT outputs are available, they can be ranked based on given score.

Several metrics are available to determine the MT quality score when one or more reference translations are available. Following four metrics are very popular and widely used in machine translation research.

- BLEU
- WER
- TER
- METEOR

2.3.1 BLEU

It determines number of n-grams of sizes varying from unigram to N-gram that the machine translation output has common with the reference translations. It then computes a modified precision measure for all the N-grams. It will give unigram precision, bigram precision to N-gram precision. Final score is calculated by taking the average *n*-gram precision over all *n*-grams up to size *N* (typically 4) using geometric mean.

$$p_n = \frac{\sum_{C \in corpus} \sum_{n\text{-gram} \in C} \text{count}_{\text{clip}}(n\text{-gram})}{\sum_{C \in corpus} \sum_{n\text{-gram} \in C} \text{count}(n\text{-gram})}$$

$$p = \sqrt[N]{\prod_{n=1}^N p_n}$$

Machine translation is a subjective issue. Different human being will produce completely different translation for a particular source sentence. So estimating recall value for machine translation output to complement the precision value is a difficult task, since we don't know which reference translation should be considered as the gold standard one. BLEU uses a penalty, called Brevity Penalty (BP), for translations that are shorter than the reference translations to compensate the recall measure. It defines the effective reference length, r , to the length of that reference translation with whom the candidate translation has the largest number of n-gram matches. Let c be the candidate sentence length.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

Final BLEU Score: $BLEU = BP \times p$

2.3.2 WER

Word error rate (WER) is a common metric of the performance of machine translation system. It is based on word level edit distance between hypothesis translation and reference translation. Minimum edit distance idea here is borrowed from Levenshtein distance, working at the word level instead of the character level. The WER is a valuable tool for comparing different systems as well as for evaluating improvements within one system. WER only consider if a particular word is present in both hypothesis translation and reference translation or not. It does not consider the relative ordering of the words. This is why WER metric can't measure smoothness of the translation in target language.

Word error rate can then be computed as: $WER = (S+D+I)/N$

Where

S is the number of substitutions

D is the number of deletions

I is the number of insertions

N is the number of words in the referenc

2.3.3 Meteor

Banerjee & Lavie (ACL-05) proposed this metric. It is based on WER metric only. WER considers only the surface form of a word. But Meteor considers stem matching and synonymy matching along with WER. If we consider post editing effort as translation metric, Meteor performs better than WER.

2.3.4 TER

Snover et al (AMTA 2006) proposed this metric. It considers word shifting along with WER. TER can be used to determine score on individual sentence level or entire file level. TER gives the translation distance between hypothesis sentence and reference sentence. Minmium TER means corresponding hypothesis sentence is better than others.

Computer Aided Translation

Statistical Machine Translation systems generate quite smooth and meaningful translation, but these translations also contain many errors. In some cases translated words are not kept in proper position, they need to be shifted to get really good translation. For some words the inflexion used may be wrong, which needs to be corrected. And sometimes a word appears in translation which should not have been appeared. Some source words may not get translated by a SMT system. Due to all these errors, it is very unlikely to use the SMT translations directly in publishing for larger public. It is always better to modify these translations by some human. This task is called Post Editing. Human post editors take help of many software systems which reduce post editing time. This type of software systems is called Computer Aided Translation system.

3.1 CAT Tools

Several Computer Aided Translation tools are available which are used by human translators in their post editing task. Some of these tools are proprietary software. Human post editors have to pay to use this software. There are some tools available which are built by researcher which is freely available for use.

3.1.1 SDL Trados

Trados GmbH developed SDL Trados⁵⁶ which is a computer aided translation software widely used in translation industry. This software is now

⁵ <http://www.translationzone.com/>

available from SDL International which is a provider of translation management software, content management and language services. It provides translation memory and terminology management. SDL Trados comes with two versions - one which is free and can be used by researcher. Another is professional edition which is used for commercial purpose.

Trados Studio 2011 has integrated machine translation and translation memory into their software suite. It tries to get a match from its translation memory. If no match found then, it uses machine translation to translate unmatched segments independently. Human post editors then can modify the output to make the translation perfect. It currently uses machine translation system like Google Translate, Weaver, and Microsoft Translator.

3.1.2 Wordfast

Yves Champollion in 1999 developed a translation memory based computer aided translation tool ‘Wordfast’⁷ as a cheaper alternative to Trados. This original product in now knows as ‘Wordfast Classic’. Presently many translation memory products are available from this company and all of them are known as ‘Wordfast’. This translation software also comes with two flavors – one which is used for commercial purpose and another which is free. Free version of Wordfast is very popular among professional translator. Some restrictions are there on free version of the software mainly how many sentences can be translated using it is limited here.

⁶ https://en.wikipedia.org/wiki/SDL_Trados

⁷ <https://www.wordfast.net/>

3.1.3 PET

PET⁸ is a stand-alone, open-source computer aided translation tool that helps to do post-edit and assess machine or human translations. It gathers detailed statistics about post-editing time, number of keystrokes required to post edit, number of edit operations required along with many other effort indicators. PET is a very useful and cheap tool to evaluate MT quality, if post editing effort is used as quality metric.

3.1.4 MateCat

MateCat⁹, acronym of Machine Translation Enhanced Computer Assisted Translation is a web-based computer-assisted translation (CAT) tool. It provides human translators a professional work environment, translation memories, glossaries, concordances, and machine translation. It represents probably the best available open source platform for investigating, integrating, and evaluating under realistic conditions the impact of new machine translation technology on human post-editing. The objective of MateCat is to improve the translation workflow by integrating machine translation (MT) and human translation within the so-called computer aided translation (CAT) framework.

3.1.5 CasMaCat

CASMACAT¹⁰ (Cognitive Analysis and Statistical Methods for Advanced Computer Aided Translation) is a web based modular workbench for computer assisted translation (CAT). The tool includes an array of innovative features to offer a rich, user focused working environment which is not available in any other CAT tool. CASMACAT works in close collaboration with the MATECAT project. However, while MATECAT is concerned with conventional CAT, CASMACAT is

⁸ <http://rgcl.wlv.ac.uk/projects/PET/>

⁹ <https://www.matecat.com/>

¹⁰ <http://www.casmacat.eu/>

focused on enhancing user interaction and facilitating the real-time involvement of human translators. In particular, CASMACAT provides highly interactive editing and logging features.

CASMACAT offers advanced functionalities for computer-aided translation and the scientific study of human translation: automatic interaction with machine translation (MT) engines and translation memories (TM) to obtain raw translations or close TM matches for conventional post-editing; interactive translation prediction based on an MT engine's search graph, detailed recording and replay of edit actions and translator's gaze (the latter via eye-tracking), and the support of e-pen as an alternative input device. The system is open source software and interfaces with multiple MT systems.

3.2 Limitations of Current CAT Tools

Existing post-editing environments have three main limitations: restricted availability and flexibility, and lack of detailed statistics from post-editing jobs. Most of them are proprietary tools only available as part of a major (and more expensive) product distribution. Apart from a few options, mostly regarding their interface, they cannot be modified in any way. Furthermore, these tools generally only allow the post-editing of one or a very small number of specific MT systems, which restricts their application. As such, they do not allow, for example, the comparison of translations produced by different MT systems in terms of post-editing effort.

CATaLog - A TM based CAT Tool

Localization is a very important task for every industrial product. Product Development Company needs to publish product details, product manuals, and user guide etc. in local languages to make it more acceptable to local people. English documents need to be translated into local languages. For first time this needs to be done manually so that quality of translation is good. But from next time most of the previous translation can be reused. In this scenario we can use TM for better productivity. Older translations will be stored in TM and they will be used as reference translation to produce desired translation of new sentences.

Even same argument for localization can be used for tourism domain or health domain. A tourist may have a limited set of queries when he or she visit some places. All these queries can be translated in local languages which will be used by local tour guides to help foreign tourists. Each time one query may not appear in same form. So problem to this little variation can be solved using TM and post editing.

Post editing tool can be built either on top of a Machine Translation (MT) system or on top of a Translation Memory (TM). But it has been noticed that MT outputs in target language are sometimes so bad that, post editor does the translation task from scratch. So MT output does not help a post editor as such. But in Translation memory reference target sentences are smooth and meaningful sentence in target language saved in database. Editors do some insertion, deletion, substitution operations on TM output to produce desired result. Statistically it has been seen that post editors prefer to use TM output much more to do post edition than MT output.

4.1 Related Research Work

There are several tools available which are developed on same idea. SDL Trados¹¹, Wordfast¹² are two such tools. These tools provide Translation Memory based output and translators use those reference translations to produce desired translation. Some of them use Machine Translation output also along with TM output. They insert machine translated phrases when TM can't find a match. One major problem with these tools is that they are mostly commercial tool and highly expensive. Some public research has been done on building CAT system too.

He et al. (2010) proposed a translation recommendation framework to integrate Statistical Machine Translation (SMT) output with Translation Memory (TM) systems. The framework recommends SMT outputs to a TM user when it predicts that SMT outputs are more suitable for post-editing than the hits provided by the TM. They described an implementation of this framework using an SVM binary classifier and exploited methods to fine-tune the classifier and investigate a variety of features of different types. Experimental results show that their system can achieve 0.85 precision at 0.89 recall, excluding exact matches. Furthermore, it is possible for the end-user to achieve a desired balance between precision and recall by adjusting confidence levels.

Espla et al. (2011) explored a new method to improve computer-aided translation (CAT) system based on translation memory (TM) by using pre-computed word alignments between the source and target segments in the translation units (TUs) of the user's TM. When a new segment is to be translated by the CAT user, their

¹¹<http://www.translationzone.com>

¹² <http://www.wordfast.com/>

approach uses the word alignments in the matching TUs to mark the words that should be changed or kept unedited to transform the proposed translation into an adequate translation. In this paper, they evaluated different sets of alignments obtained by using GIZA++. Experiments conducted in the translation of Spanish texts into English show that this approach is able to predict which target words had to be changed or kept unedited with accuracy above 94% for fuzzy-match scores greater or equal to 60%. In an appendix they evaluated their approach when new TUs (not seen during the computation of the word-alignment models) are used.

Work on integration of Machine Translation system with Translation Memory (Kanavos and Kartsaklis, 2010) has also been done to make TM output more meaningful. They chose a fuzzy cut off 80% to decide whether to accept a TM output or not. If TM matches is more than 80% then post edit the TM output itself. If it is less than 80%, then using classifier decide whether to accept and edit this TM output or to reject it. If it is rejected then use MT to give reference translation. Otherwise use MT system to insert mismatched phrases in TM output.

Koehn and Senellart (2010) worked on convergence of Statistical Machine Translation and Translation Memory. Test sentence is matched with reference sentences stored in TM. Matched portion of test sentences are kept as it is. Mismatched part of a TM output is replaced with phrases matched by MT system. It has been seen that this type of system performs better than standalone MT or TM system.

Dandapat et al. (2012) showed that hybridization of EBMT and SMT system produces better result than just SMT. They compared result of an EBMT system with significant amount of TM and hybrid system of EBMT and SMT. Hybrid system produces significant improvement in translation quality for English-Turkey and

English-French translation. For large TM, EBMT system suffered from time complexity issue. It was taking long to determine similarity score between test sentence and TM sentences. So they used length based heuristic and IR based indexing to reduce the comparison set. They were only comparing those sentences in TM with test sentence which have similar length. That prunes many TM sentences. IR based indexing also help in pruning many TM sentences.

Wang et al. (2013) proposed integrated model to incorporate statistical machine translation (SMT) and translation memory (TM), since they complement each other in matched and unmatched regions. Unlike previous multi-stage pipeline approaches, which directly merge TM result into the final output, the proposed models refer to the corresponding TM information associated with each phrase at SMT decoding.

4.2 System Description

We have built English to Bengali TM system. We have English to Bengali parallel corpora storing English sentences and their Bengali translation in separate files. A test sentence will be matched against these stored sentences and they will be ranked according to their similarity with the test sentence. Top 5 most similar sentences will be chosen. We have used Translation Error Rate based metric to rank the reference sentences. Each choice will include both English sentence and their corresponding Bengali translation. We align the test sentence with the referenced sentences and then align each reference sentence with their translation. From these two alignments we try to find which part of translation may matches with the actual translation of test sentence and which does not match. Matched parts and unmatched parts will be color coded for each choice. Matched parts will be given green color and Unmatched parts will be given red color. Seeing the amount of green and red portion in a translation post editor can choose which one out of those 5 will produce desired

translation with minimum effort. User can edit the sentences on the tool and can save the actual translation.

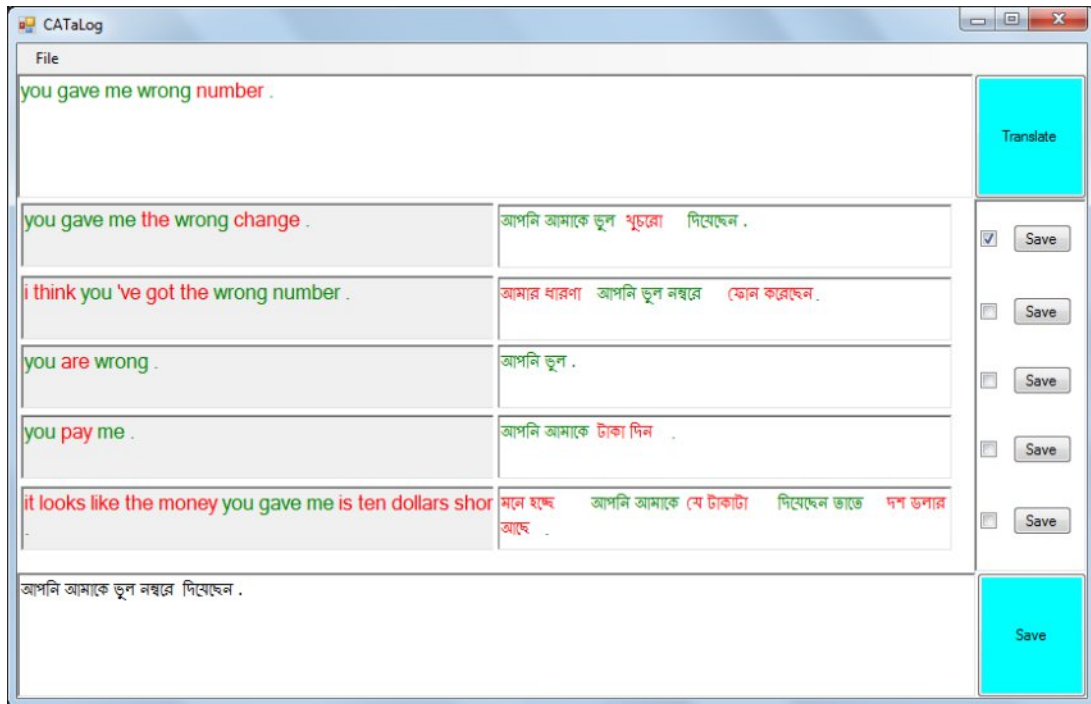


Figure 1 : CATaLog Main Screen

4.3 Corpus

Our training corpus contains more than 10000 English sentences and their Bengali translation. We have kept English sentences in one file named “train.en” and its corresponding translation in another file named “train.bn”. Sentence ending delimiter is ‘.’ for both English and Bengali sentences. We have used two test files- one containing 100 English sentences and other contains 500 English test sentences.

We have generated alignment between training source sentences and target sentences using Moses¹³ and GIZA++¹⁴. This alignment file is used to generate color coding between selected source and target sentences.

Following is an example how we store our training corpus.

English Sentence: we want to have a table near the window .

Bengali Translation: আমরা জানালার কাছে একটা টেবিল চাই ।

Alignment: NULL ({ }) we ({ 1 }) want ({ 6 }) to ({ }) have ({ }) a ({ 4 }) table ({ 5 }) near ({ 3 }) the ({ }) window ({ 2 }) . ({ 7 })

Alternatively bi-lingual alignment can be present as follows.

1-1 2-6 5-4 6-5 7-3 9-2 10-7

4.4 Similarity Metric

tercom-7.25¹⁵ is used to determine distance score or translation error rate (TER) between test sentence and the reference sentences which are in same language. Also it gives the alignment between them. Since we are ranking reference sentences based on their similarity with the test sentence, we can use the TER score in inverse way. Lower the TER score means, higher the similarity. TER gives a real value score to convert one sentence to another sentence in same language. We can directly use the TER score for ranking of sentences. But in our system we have used our own score calculation technique based on the alignment provided by TER. Actually TER gives equal weight to deletion, insertion and substitution operation. But in post editing deletion takes much lesser time than insertion or substitution. Different cost for each operation produce better result. These weights can be adjusted to get better output from TM. We have set match_reward=0.0, deletion_cost=0.20, insertion_cost=0.50, substitution_cost=0.70 and shift_cost=1.0 to get TER alignment. After getting the

¹³ <http://www.statmt.org/moses/>

¹⁴ <http://www.statmt.org/moses/giza/GIZA++.html>

¹⁵ <http://www.cs.umd.edu/~snover/tercom/>

TER alignment we calculate the similarity score to rank the TM sentences. Following example explains our scoring technique well.

Input : you gave me wrong number .

Suggestion : you gave me the wrong change .

TER Alignment : M M M D M S M

We set following reward values and cost values to calculate the similarity score.

Reward (M) =0.90, cost (I) =0.50, cost (D) =0.20, cost (S) =0.70

Similarity score = $5*0.90-1*0.20-1*0.70 = 3.6$

We keep all the cost and reward values between 0 and 1. We select 5 sentences which have maximum similarity score and their corresponding translation.

4.5 Color Coding

Among the top choices, post editor will select one reference translation to do the post editing task. To make that decision process easy, we color coded the matched part and unmatched of each reference translation. Green portion implies that they are matched fragments and Red portion implies mismatch.

GIZA++ generates the alignment between training source sentence and target sentence. This alignment file is generated once on the training data, if new training data is not added. TER gives us the alignment between test sentence and selected sentences. Using these two alignments we give color-green to the matched phrases and color-red to the unmatched phrases of selected source sentence and its corresponding translation. It indicates which portion of selected translated sentence may have matched with input/test sentence and which are not.

A reference translation which has almost same length with the test sentence, and has more green fragments than red fragments will be a good candidate for post editing.

Sometimes smaller sentences may get near 100% green color, but they are not good candidate for post editing, since post editors may have to insert more words. Insertion is costly operation in post editing.

Following is an example output of our system.

Input Sentence: you gave me wrong number .

English Match:

1. you gave me the wrong change .
2. i think you 've got the wrong number .
3. you are wrong .
4. you pay me .
5. you 're overcharging me .

Bengali Translation:

1. আপনি আমাকে ভুল খুচরো দিয়েছেন .
2. আমার ধারণা আপনি ভুল ন?রে ফোন করেছেন .
3. আপনি ভুল .
4. আপনি আমাকে টাকা দিন .
5. আপনি আমার কাছে থেকে বেশি নি?ন .

4.6 Improving TM Search Time

Comparing each input/test sentence with more than 10000 training sentences makes the TM very slow. In practical scenario to get good result from a TM training corpus may be much bigger than this. In that case determining TER score will take lot time for all reference sentences. We have used concept of inverted index and length based pruning to make TM search faster.

4.6.1 Inverted Index

Concept of inverted index used in most search engine can improve the TM search time significantly. On training data we create vocabulary list after removing stop

words and other tokens which have no importance to determine similarity. We store all vocabulary in lower case. Though this has a major drawback, but for simplicity we have used this idea. We don't do any stemming of words. We want to store the words in their surface form, so that if they appear in the same form in some input sentence, we will get exact meaning. For each word in the vocabulary we maintain a posting list of sentences which contain that word.

We only consider those training sentences for similarity measurement which contain one or more vocabulary word of input/test sentence. This will reduce the size of comparison set of reference sentences and the time taken to produce the TM output. Tool provides option to use these postings or not. This feature is there to compare results using postings and without using postings. In ideal scenario TM output for both should be same, though time taken to produce the output will be significantly different.

Following is format of storing postings.

pacific: 21|7505|

ocean: 21|1739|7505|7875|

Input: you gave me wrong number .

you: 3, 8, 9, 15.....

gave: 4, 8, 16.....

want: 1, 2, 5, 7, 10, 12.....

me: 4, 9.....

wrong: 2, 9, 20.....

number: 3, 17....

Comparison set: 2, 3, 4, 8, 9, 15, 16, 17, 20.....

We have pruned sentence 1, 5, 7, 10....

4.6.2 Length Based Pruning

To improve translation memory search time, we pruned some sentences using length of sentences. Sometimes smaller sentences in corpus may get total match with some part of input sentence. These sentences will get 100% green color. But since more number of words needs to be inserted in these sentences, they may not be proper candidate for post editing. We can prune some sentences whose lengths are much smaller than the input sentence. Idea is to compare test sentences with those sentences in translation memory which has comparable length to test sentences. We are considering only word level sentence length. Sometimes character level sentence length may be useful, but we are not considering it.

Input : you gave me wrong number .

Sentence 1: you gave .

Sentence 2: you gave me wrong change .

Though first sentence has 100% match, post editor needs to insert many new words to get proper translation. But in second one, only one substitution needed. That's the reason length based pruning can help to reduce corpus size.

4.7 Bulk Translation Facility

This tool gives option to translate sentences in bulk mode. Post editor can give the source path and then generate TM output at a time. This output will include the tag like <D> for deletion, <S> for substitution, <I> for insertion for each word. If there is continuous insertion or continuous deletion or continuous substitution then rather than putting those tag for each word, it create tag like xml. This output will be helpful to post editor to do the task. Since bulk translation can take too long time, tool provides option to stop translation any time in between. Because of this xml

type of tags this bulk translation file can be parsed later by any application and color coded style can be generated.

Following is format of bulk translation.

Input: we want to have a table near the window .

1: we <D>'d</D> <S>like</S> to have a table near the window .

---> আমরা জানালার কাছে একটা টেবিল <S>চাই</S> ।

2: we <S>'d like</S> a table near the window .

---> আমরা জানালার কাছে একটা টেবিল <S>চাই</S> ।

3: we <D>'d</D> <S>like</S> to have a table near the <S>street</S> .

---> আমরা <S>রাপার</S> কাছে একটা টেবিল পেতে <S>চাই</S> ।

4: we <S>would like</S> a table <S>by</S> the window .

---> আমরা জানালার ধারে একটা টেবিল <S>চাই</S> ।

5: <D>any chance</D> we <D>could</D> have a table near the window <S>?</S>

---> জানালার পাশে আমরা একটা টেবিল <D>পাবার কোনো স?বনা</D> আছে <S>?</S>

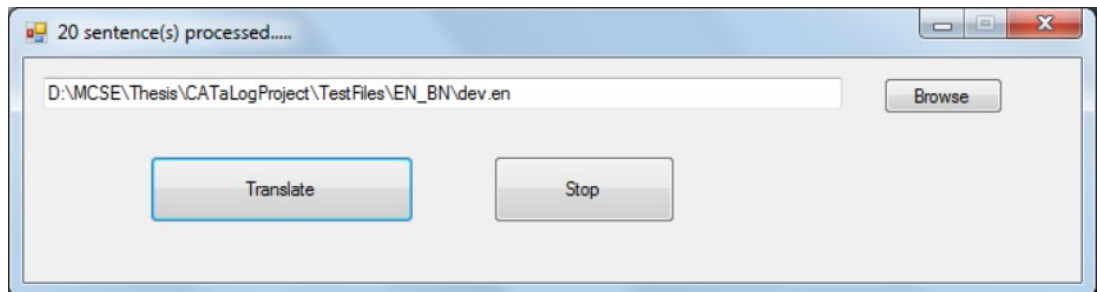
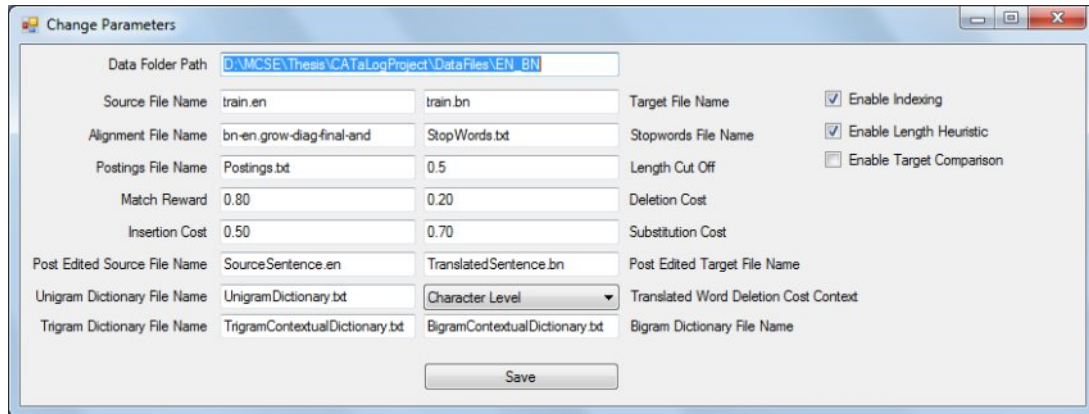


Figure 2 : Bulk Translation Form

4.8 Controlling Application Parameters

Different parameters used in this tool like TM corpus path, cost of different post editing operations can be configured from GUI itself. These parameters can be changed even in runtime, and new values will be applied in runtime itself. Last used parameters will be saved and when the tool will be launched next time, those last

used parameters will be retrieved from saved location. We can set if we want to use different TM search efficient technique like inverted index or length based pruning.



Parameter	Value	Parameter	Value	Option
Data Folder Path	D:\MCSE\Thesis\CATaLogProject\DataFiles\EN_BN	Target File Name		<input checked="" type="checkbox"/> Enable Indexing
Source File Name	train.en	train.bn		<input checked="" type="checkbox"/> Enable Length Heuristic
Alignment File Name	bn-en.grow-diag-final-and	StopWords.txt		<input type="checkbox"/> Enable Target Comparison
Postings File Name	Postings.bt	0.5		
Match Reward	0.80	0.20		
Insertion Cost	0.50	0.70		
Post Edited Source File Name	SourceSentence.en	TranslatedSentence.bn		
Unigram Dictionary File Name	UnigramDictionary.txt	Character Level		
Trigram Dictionary File Name	TrigramContextualDictionary.txt	BigramContextualDictionary.txt		

Save

Figure 3 : Parameter Settings Form

Mismatched Segments (MS) Fusion in CATaLog

This chapter explores how the translations of the unmatched parts of the input sentence could be discovered and inserted into the Translation Memory (TM) suggestions generated in a Computer Aided Translation (CAT) tool using parse tree and parts of speech tags to form a new translation which is more suitable for post-editing. CATaLog (Nayek et al., 2015) is a CAT tool based on Translation memory and modified Translation Error Rate (TER) (Snover et al., 2006) metric. Unmatched parts of TM suggestions can often be found in some other TM suggestions or in sentences which are not part of TM suggestions. Therefore, we can find the translations of those unmatched parts within TM database itself. If we can merge the translations of the unmatched parts into one single sentence in a meaningful way, then post-editing effort will be much less. Inserting the translations for the unmatched parts into TM suggestions may lead to loss of fluency in the generated target sentence. To avoid that, we use parsing and POS tagging together with a back off POS n-gram model to generate new translation suggestion.

Computer-aided translation tools (CAT) are most widely used by many language service providers, freelance translators to improve translation quality and to increase translator's productivity. CATaLog is such a CAT tool developed based on Translation Memory (TM). CATaLog uses modified TER as the similarity metric. It introduced the concept of color-coding the TM suggestions both in the source language and the target language. Matched and unmatched parts are color coded in green and red color, respectively, to facilitate post-editing and to guide the user. The intuition behind the color coding scheme in CATaLog is that the more the green color in a TM suggestion, the more the matching and hence less post editing effort is

required. It also provides options for length based sentence pruning and indexing technique to minimize the search time. CATaLog has been specifically designed to improve user experience with TM. In this paper, we report additional functionality to the CATaLog tool and TM technology in general.

TM tools traditionally do not generate any translation; instead they present the user with matching sentence pairs that are very similar to the sentence being translated. Post-editors, when working with TM tools, seldom find an exact match. Therefore, almost all the times, the TM suggestions contain at least a few unmatched parts. However, it can often be observed that the translation for those unmatched parts are available in other suggestions or may be in some other sentences in the TM database. If we can extract the translations for those unmatched parts from other sentences and introduce them into the suggested TM translations, then we can generate almost the entire translation for a particular input sentence. While filling those unmatched parts may lead to loss of fluency in the suggested new translation, it improves the accuracy of the suggested translation. Thus, it reduces the post-editing cost significantly since the user does not have to type in the entire translation for the unmatched parts. We introduced this new capability to the CATaLog tool which is reported in this paper.

5.1 Related Work

CAT tools are very popular among professional translators. They use these tools in their translation workflows in a regular basis to reduce translation time and improve productivity. Along with basic research on CAT tools, some researchers tried to fill the gaps for the mismatched parts in the input sentence in different ways.

Biccici and Dymetman (2008) combined dynamically extracted source-target phrase pairs from the TM with the phrase table of a phrase-based SMT (PB-SMT) system. Translation for a mismatched part is taken from this phrase table and the translation is replaced in the target sentence.

Simard and Isabelle (2009) combined TM with PB-SMT to enable PB-SMT to take advantage of exact or fuzzy matching features of TM. They proposed two different strategies: (i) an MT-TM combined system where, above a certain similarity threshold value, the combined system provides the translation from the TM, otherwise it produces the MT output; and, (ii) it allows the PB-SMT system to actively exploit the most similar material identified by the TM, via TM-based feature functions.

Zhechev and Van Genabith (2010) explored similar strategies, but uses syntactic information during fuzzy matching by applying sub-tree alignment in order to link nodes between the input sentence, TM match and TM translation. Sub-tree based alignments reliably determine the correspondences between an input sentence and a TM suggestion. An SMT system is used to translate the mismatched parts of the input sentence. The complete translation ensures higher quality than the TM suggestions.

Koehn and Senelart (2010) proposed a similar method to combine MT with TM. They used fuzzy matching to retrieve similar segments from the TM for each source segment that needs to be translated and identified the mismatched parts using automatic word alignment. Finally, those mismatched parts are replaced by SMT translations.

Ma et al. (2011) uses support vector machine and discriminative learning method to identify the matched words to select a translation unit. They addressed several problems in fuzzy matching based translation unit identification. In general, translation units with lower fuzzy match value are thrown away, however, their method considers those units during translation.

Dandapat et al. (2011) also worked on identifying unmatched parts of the input sentence and replacing their translations in the TM candidate translation. From the original translation memory, they first identified sub-segment level translation pairs which form a sub-segment TM. When some unmatched sub-segment is found in the input sentence, they look for its translation in the sub-segment TM. They did not consider the context of the unmatched sub-segments while inserting their translations; they just plugged those sub-segments translation into the target sentence based on how they appear in the input sentence.

5.2 System Description

CATaLog generates top 5 suggestions based on modified TER. Whenever the post-editor chooses one suggestion for post-editing, the CAT system tries to fill up the unmatched parts of that suggestion and presents the user with a new translation suggestion. The system components are detailed in the following sections.

5.3 Generating Dictionary

We tried to fill the unmatched parts of a TM suggestion at word level. Whenever we find some unmatched words in the input sentence to be translated, we need to look for their translation(s) somewhere. One way of achieving this is to keep a bilingual dictionary. However, dictionary is a costly resource for many language pairs. Therefore, rather than using a built-in dictionary, we generate a dictionary from the background bilingual corpus available with the translation memory. For illustration

purpose, all the examples presented in this paper are in English--Bengali which are obtained from an English--Bengali parallel corpus with 13,000 sentences. English is considered as the source language and Bengali as the target language in the present work. We generate English to Bengali dictionary from the parallel corpus where English words are stored along with their parts of speech information and their corresponding translations in Bengali. In the present work, we used the Stanford POS tagger¹⁶ to generate the POS tags for the source side of the parallel corpus. GIZA++ (Och and Ney, 2003) implementation of the IBM word alignment model (Brown et al., 1993) is used to produce one to many alignments between source and target language words. From this source--target alignments we find out the translation correspondences of each English word available in the parallel corpus. This dictionary is generated offline for once and it gets loaded when the TM application is loaded.

The meaning of a polysemous word depends on the context it appears in. In case of translation, a source word can have completely different translation or may have different suffixes attached to it based on its context. Therefore, to determine which translation is more accurate in a particular context, we look at the neighboring context. Instead of considering the lexical context, we take into consideration the parts of speech (POS) context in this work. In our current system, we use a trigram back-off model for determining contextual translation of a source word. We generate three dictionaries: one is trigram context based, second one is bigram context based and the third one is simply a unigram dictionary. Here context refers to parts of speech sequence context. In trigram contextual dictionary, for a particular source word, we store the previous two POS tags, POS tag of the word under consideration and the next two POS tags. We also store the frequency of this entire context tag

¹⁶ <http://nlp.stanford.edu/software/tagger.shtml>

sequence (in the training corpus) along with the particular translation for that word. Trigram context based dictionary entries look like as follows:

bottle: VBP_DT_NN_IN_NN|2|একবোতল|1; PRP_CD_NN_IN_NN|2|বোতল|1;

The example given above is the dictionary entry corresponding to the word 'bottle'. Here the POS tag sequence is VBP_DT_NN_IN_NN. Number '2' represents the zero based positional index of the POS tag of the word. 'এক বোতল' is the corresponding translation. Number '1' appearing at the end, represents the frequency of this translation in the training corpus for the word 'bottle' for this particular POS context.

We follow the same format for all the 3 dictionaries. The entries in the bigram and unigram context based dictionaries corresponding to the word 'bottle' are given below.

bottle: DT_NN_IN|1| একবোতল|1; CD_NN_IN|1|বোতল|1;

bottle: NN|বোতল|7; NN| একবোতল|6; NN|বোতল খুলে|1; NN|বোতল||1;

In order to find the translation of a non-matching word in the input sentence, we first look at the trigram dictionary; if no match is found there, we look for a match in the bigram dictionary and finally back-off to the unigram dictionary. If multiple matches are found in any particular dictionary, we choose the most frequent translation from among them. In case of a frequency tie, which is very unlikely, we choose any one of them randomly. While doing the POS context matching, we first try to get an absolute match first. If there is no absolute POS context match, then we look for approximate POS context match. This concept of absolute POS context match and approximate POS context match is explained in next section.

5.4 Grouping Parts Of Speech Tags

We used the Stanford POS tagger to generate the POS tag sequence for the input sentence. Use of this POS tag is explained in later section of the paper. The Stanford tagger uses the Penn Treebank tagset¹⁷ which contains 38 different types of POS tags. However, since we intend to generate this translation for use in post-editing, we relax the constraint of exact POS tag match. Therefore, we group together similar POS tags into coarse grained POS categories to get more matches between the input sentence and the TM suggestions. E.g., we group together VB, VBD, VBZ, VBN, VBG, and VBP into a coarse grained group called VB (i.e., verb). Similarly we group JJ, JJR, and JJS into a JJ group. NN, NNS, NNP, NNPS are grouped into the NN group. POS tags like RB, RBS, and RBR are grouped into the RB group. When performing POS tag matching between the input sentence and TM suggestion, first we look for an absolute match, i.e., POS tag match. In case of no absolute match we go for matching at basic POS category level.

5.5 Finding Translations for Unmatched Parts

Since CATaLog uses the TER metric as the measure of similarity between the input sentence and the TM database, as a byproduct, TER also provides the alignment between input sentence and selected TM suggestion sentences. From this alignment we can easily find out which words of the input sentence do not match with the suggestion sentence.

Input Sentence: you gave me wrong number .

TM Suggestion: you gave me the wrong change .

TER Alignment: M M M D M S M

Here 'M', 'D', 'I' and 'S' correspond to match, deletion, insertion and substitution operations, respectively.

¹⁷ https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

`I' and `S' editing operations in TER alignment correspond to the unmatched words between the input sentence and the corresponding TM suggestion. `D' editing operation corresponds to deletion of extra words that are not present in the input sentence and this editing operation is easily catered to by simply deleting the corresponding word(s) in the target identified through word alignments. In CATaLog tool these words are shown in red color. `I' signifies that the corresponding input sentence word has to be inserted, while an `S' represents the substitution of the TM suggestion word for the input sentence word. In either case, the translation of the unmatched input sentence word has to be inserted to generate the translation of the input sentence.

If we do not consider the POS usage of the unmatched word in the input sentence, we could only use the most frequent translation of that word from the dictionary. In that case we also do not need to store the POS information in the dictionary. However, considering the POS usage of the unmatched word allows us to be more specific about its translation in the context. E.g., the word `book' has the following entry in dictionary.

book: NN|বই; VBD|সংরক্ষণ করা

If the unmatched word `book' in input sentence is used as NN then the system chooses the translation ``বই". If it is used as VBD, then the system prefers the translation ``সংরক্ষণ করা". If the word cannot be found in the dictionary then it remains un-translated. While matching the POS tag we might not find the exact POS tag match. In that case we try to find an approximate POS match, i.e., at the coarse-grained POS category level. For example, if the `book|VBD' word|POS combination cannot be found in the dictionary, we then look for the dictionary entry for the word `book' together with POS category VB, i.e., any of VB, VBD, VBZ, VBN, VBG, and VBP. If, say for example, the `book|VBZ' combination is found, the

corresponding translation is taken. The notion is that the translation corresponding to `book|VBD' might not be exactly the same as `book|VBZ'; however, they are derived from the same root verb and thus a little post-editing should result in the right translation. The POS based dictionary matching thus reduces the POS level ambiguity.

5.6 Finding Positions to Insert Translations

After obtaining the translations of all the unmatched words, we need to find out where to put these target language words in the selected TM suggestion translation. If these target language words are not put in an appropriate manner, then the suggested new translation becomes less fluent and ineligible for post editing. TM, despite being technologically very simple, has proved itself to be a widely used technology in the localization industry mainly because of its strength that it presents the user with perfectly fluent translation suggestions for post-editing. Thus, presenting the user with a more accurate but less fluent translation suggestion might not be acceptable. Our idea is to use POS tagging and parsing to guide the identification of the proper location for insertion of the translation of the unmatched word. To achieve this we subject both the input sentence and the selected suggestion sentence to POS tagging and parsing. POS tagging and parsing are performed in the present work using the Stanford POS tagger¹⁸ and Stanford parser¹⁹, respectively.

5.7 Matching using POS N-grams

When we search for the location for inserting the translation of an unmatched word, we first try to find a corresponding word in TM suggestion sentence that does not match with any word in the input sentence. Successively, we find the words and their positions in the target side of the parallel sentence that the unmatched source

¹⁸ <http://nlp.stanford.edu/software/tagger.shtml>

¹⁹ <http://nlp.stanford.edu/software/lex-parser.shtml>

word corresponds to. Those positions are the potential positions where the translation of the unmatched word can be put. The corresponding word in the TM suggestion sentence can be found using POS tag of the unmatched word. We try to find the same POS tag in source suggestion sentences and the corresponding word should not match with any other word in input sentence. Once we find such a corresponding word, we mark it so that the next time when we try to find another corresponding word for another unmatched word we do not consider it again. While matching the POS tag, first we consider the absolute POS tag match. If we do not find any such match, we go for finding a POS tag which belongs to same POS category as described in the previous section.

We have used a back off POS trigram model for searching the location of the corresponding word. In this model, an n-gram represents a sequence of three consecutive POS tags. Since we consider trigram POS sequences, we have to take into consideration three different trigrams. We test each of the three POS trigrams individually. If none of them matches with any POS trigram sequence in the selected suggestion translation, then we go for back off POS bigram matching. If multiple matches are found then we resolve the ambiguity using parse tree information of the input sentence to determine which trigram sequence is more suitable. This parse tree matching process is detailed in the following subsection. If we do not find any higher order n-gram match, we fall back to unigrams. If the system fails to find even a unigram POS tag match (i.e., word|POS), then the unmatched word in the input sentence remains un-translated. For such words, the system disregards the POS of the word and provides a drop down list of probable translations which becomes available on right click of the mouse from which the user can directly choose a translation (as opposed to typing) by left click of the mouse and can put the target word in a proper place.

5.8 Parse Tree Matching

When multiple POS n-gram matches are found, the system resolves this ambiguity using the parse tree of the input sentence. For all the higher order POS n-gram matches, we determine the lowest common ancestor node in the parse tree. The n-gram POS sequence choice for which the depth of the common ancestor node is maximum is considered as the winner. If there is a tie, then the system chooses one among them randomly. The idea behind choosing the lowest common (i.e., maximum depth) ancestor is that the lower the common ancestor in the parse tree, the more they are related. If the lowest common ancestor is located at the top of the sub-tree, the words considered in the n-gram sequence are unrelated and hence they should be ignored. This motivates our philosophy behind using the lowest common ancestor.

After we have found the location of the corresponding word in the selected TM suggestion, we determine the positions of translation of that word in the translation of that TM suggestion using the alignment generated by GIZA++. These positions in TM suggestion translation are the potential positions where the translation of the unmatched word could be to put. Since GIZA++ generates one to many alignment from source to target translation, three situations can arise here. The length of the translation of the corresponding word could be equal to, or shorter, or longer than the length of the translation of the unmatched word.

Let w_1 be the unmatched word in the input sentence and w_2 is the corresponding word in a TM suggestion. The length of the translation of w_1 could be equal to, shorter, or longer than the length of the translation of w_2 . We define the length in terms of number of words. Translation of the word w_2 may be one word or multi word. In case of multiword meaning of w_2 , those words may be continuous or non-continuous in the translation. That means the potential positions for inserting the

translation of w_1 also may be continuous or discontinuous in the TM suggestion translation. If the translation of w_2 has the same length or is longer than the translation of w_1 , then we just replace the translation of w_1 in those positions. We replace one word in translation of w_2 with one word from the translation of w_1 . Therefore, our system will work properly even if the potential positions of insertion are not continuous. Some positions will not be replaced in case the translation of w_2 is longer than the translation of w_1 . However, if the translation of w_1 is longer, we merge the extra words of the translation of w_1 with its last word (separated by space) and put this merged word in the last position of translation of w_2 . In this way we place the translation of unmatched word(s) of the input sentence in the TM translation suggestion.

5.9 Illustration with an Example

The process is illustrated below with two examples. In the following two examples, for the sake of simplicity, we just make use of the unigram dictionary to get the translation for the unmatched words. However, in the actual system, a trigram back-off model is used.

Input sentence: i would prefer something in a middle price range .

TM suggestion: i would prefer to sit in the back part of the plane .

TER alignment: M M M D S M D D S S S S M

TM suggestion translation: আমি বিমানের পিছনের অংশে বসতে পছন্দ করব .

Table 1 : Test Sentence and TM Suggestion Alignment

Input	TM Source	Alignment	TM Translation
I	I	M	আমি
would	would	M	
prefer	prefer	M	পছ? করব
	to	D	
something	sit	S	বসতে
in	in	M	
	the	D	
	back	D	পিছনের
a	part	S	অংশে
middle	of	S	
price	the	S	
range	plane	S	বিমানের
.	.	M	.

The unmatched words in the input sentence in this case are 'something', 'a', 'middle', 'price', and 'range'

Unigram dictionary entries for the above mentioned unmatched words are:

something: NN|একটা কিছু; NN|কিছু; NN|কোন কিছু; NN|কিছু একটা

a: DT|একটা; DT|কোন; DT|এক

middle: JJ|মাঝারি আকারের; JJ|মাঝের

price: NN|দাম; NN|দামটা; NN|মূল?

range: VBP|দেড়শ এর মধ্যে বদলাতে থাকে

POS sequence for the input sentence: i/FW would/MD prefer/VB something/NN in/IN a/DT middle/JJ price/NN range/NN ./.

POS sequence for TM suggestion : i/FW would/MD prefer/VB to/TO sit/VB in/IN the/DT back/JJ part/NN of/IN the/DT plane/NN ./.

The CAT System searches for matching translation examples for those unmatched words in the same context as they appear in the input sentence. Here that sequence is 'something', 'a', 'middle', 'price', and 'range'.

For the word 'something/NN', the 3 trigram sequences used for search are: 'would/MD prefer/VB something/NN'; 'prefer/VB something/NN in/IN'; 'something/NN in/IN a/DT'.

System found a match with third trigram sequence in TM suggestion. The match sequence is 'part/NN of/IN the/DT' where the word 'part/NN' does not match with any word of input sentence. So system does not go for bigram or unigram matching search. Now system go for searching the position where meaning of 'part' is located in TM translation. For that it's use GIZA++ alignment. Following is the GIZA++ alignment for TM suggestion:

1-1, 3-6, 3-7, 5-5, 8-3, 9-4, 12-2, 13-8

Here position index before hyphen (-) is the word position in TM suggestion in source language and position index after hyphen (-) is the word position in TM

suggestion translation. 'part' is 9th word in TM source suggestion. So its meaning is the 4th word in translation which is 'অংশে'. This 'অংশে' will be replaced by 'একটা কিছু'. Now suggestion translation will be:

আমি বিমানের পিছনের একটা কিছু বসতে পছন্দ করব.

Next, system will go for searching the match of 'a/DT'. Trigram POS sequences are 'something/NN in/IN a/DT'; 'in/IN a/DT middle/JJ'; 'a/DT middle/JJ price/NN'. Here 'part/NN of/IN the/DT' sequence starting with part/NN has already matched with 'something'. So this match will not be considered again. But we will get match of other two trigram sequence with 'in/IN the/DT back/JJ' and 'the/DT back/JJ part/NN' where 'the/DT' has not matched with any word of input sentence. Now to resolve the ambiguity we need to consider the parse tree of input sentence. Parse tree of input sentence is shown below.

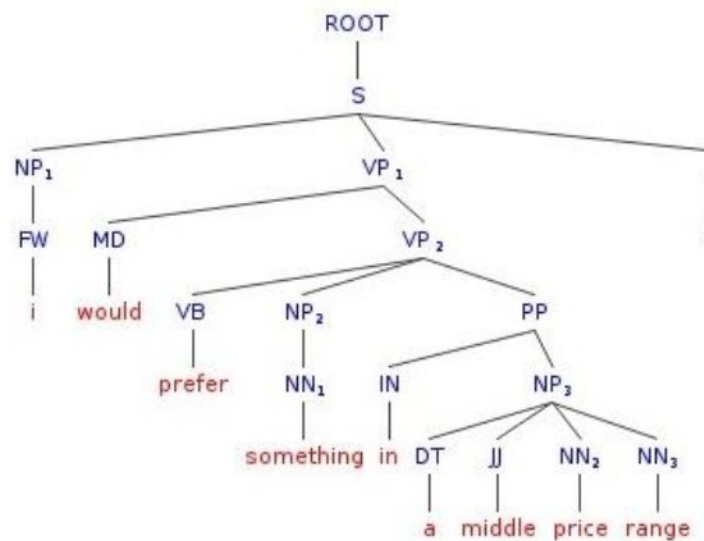


Figure 4 : Parse Tree

'In/IN a/DT middle/JJ' has lowest common ancestor at depth 4. But 'a/DT middle/JJ price/NN' has lowest common ancestor at depth 5. So we should consider this trigram. So the matched sequence is 'the/DT back/JJ part/NN'. So system will look

for meaning of 7th word 'the' of TM suggestion. But there is no match in GIZA++ alignment. So the meaning of 'a' will not be put in TM suggestion translation.

System will now search for 'middle/JJ'. Trigram sequences are 'in/IN a/DT middle/JJ'; 'a/DT middle/JJ price/NN'; 'middle/JJ price/NN range/NN'. First two trigram will be matched with 'in/IN the/DT back/JJ' and 'the/DT back/JJ part/NN' where 'back/JJ' has not matched with any word of input sentence. Third trigram will not be matched with any POS sequence in TM suggestion. To resolve ambiguity we need to consult the parse tree again. Trigram POS sequence 'in/IN a/DT middle/JJ' has lowest common ancestor at depth 4 and trigram POS sequence 'a/DT middle/JJ price/NN' has lowest common ancestor at depth 5. So this trigram will be considered and corresponding word for 'middle/JJ' is 'back/JJ'. 'back/JJ' is located at position 8 of TM suggestion and its translation is 'পিছনের' which is located at position 3 of TM suggestion translation. So meaning of 'middle/JJ' 'মাঝারি আকারের' will replace the word 'পিছনের' in TM suggestion translation. So now the modified translation will look as follows:

আমি বিমানের মাঝারি আকারের একটা কিছু বসতে পছন্দ করব .

System will now search for 'price/NN' whose meaning to be used here is 'দাম'. Trigram sequences are 'a/DT middle/JJ price/NN'; 'middle/JJ price/NN range/NN'; 'price/NN range/NN ./.' . Here 'a/DT middle/JJ price/NN' will get a match with 'the/DT back/JJ part/NN', where 'part/NN' is the corresponding word for 'price/NN'. But 'part/NN' has already been used earlier. So system will ignore this match. Other two trigrams will not be matched with any POS sequence of TM suggestion. Bigram sequences for 'price/NN' are 'middle/JJ price/NN' and 'price/NN range/NN'. 'middle/JJ price/NN' will be matched with 'back/JJ part/NN', but it will be ignored since meaning position of 'part/NN' already been replaced. Other bigram will not be matched. So system will now search for unigram 'price/NN' match. It will be

matched with `part/NN' and `plane/NN'. But `part/NN' has already been used. So system will consider `plane/NN' which is at position 12 of input sentence and its meaning `বিমানের' is at position 2 of suggestion translation. So this word `বিমানের' will be replaced by `দাম'. So modified translation will be:

আমি দাম মাঝারি আকারের একটা কিছু বসতে পছন্দ? করব .

Now system will go to find a match for `range/NN'. But its trigram, bigram, and unigram POS sequences are either being used already or does not match. So its meaning will not be put in suggested translation. Finally word `বসতে' which is translation of `sit' will be deleted since `sit' does not match with any word of input sentence. So final translation suggestion will be:

আমি দাম মাঝারি আকারের একটা কিছু পছন্দ? করব .

Since the meaning of `a/DT' and `range/NN' is not replaced in translation suggestion, their meanings `একটা' and `দেড়শ এর মধ্যে বদলাতে থাকে' will be added to a list and will be shown to post editor as suggestion. Post editor can directly use those meaning without typing them and can put them in proper place. In this way the system modifies the TM translation suggestion to generate more appropriate translation candidates. These translation candidates can be post-edited with less post-editing effort.

5.10 Length Based Pruning

The POS tags and parse tree based process of filling up of the translation of the unmatched word in the TM translation suggestion to make it more suitable for post-editing works well when the input sentence and the suggestion translations are of similar lengths. If the input sentence and suggestion sentence has completely different length, then their parse tree will be completely different. In such cases looking for POS sequence match involving the unmatched word in the suggestion sentence can give us wrong results, which will eventually lead to loss of fluency in

the target translation. Therefore, we consider only those sentences for translation suggestion whose lengths are within a specified limit of the length of the input sentence. Sentences which are either too short or too long with respect to the input sentence are pruned. This reduces the time to generate initial translation suggestion also.

5.11 Re-ranking of TM Suggestions

Our experiments, which is discussed in next section, shows that if we choose the best suggestion from the top suggestions provided by CATaLog_MS system than the first one, we are getting higher BLEU score and lower TER value. Here suggestions are ranked based on S-BLEU score. This result forced us to do re-ranking of suggestion translation so that the best one appears at the first suggestion. We have used language model, length of test sentence and length of source side suggestion sentence, number of unmatched words meaning successfully inserted in the translation of suggestion along with original similarity score of CATaLog system to re-rank the suggestions. Result of this system is included in next section as CATaLog_MS_ReRank system. This results shows that if we choose the first suggestion after re-ranking, we have achieved better BLEU score and TER value than the Best suggestion of CATaLog_MS system.

Original CATaLog score is calculated based on TER alignment.

Let's assume

match_reward=0.80,

deletion_cost=0.20,

insertion_cost=0.50,

substitution_cost=0.70,

shift_cost=1.0

Then if TER alignment is like MMDIMISMM then

$$\text{OriginalTMMatchScore}=0.80*5-0.20-0.50*2-0.70=2.1$$

Let's say CATaLog_MS system successfully insert meaning of two word represented by 'I' in TER alignment. Then we should add match_reward for these two words in original score.

$$\text{NewTMMatchScore}=2.1+2*0.80=3.7$$

We then estimate the smoothness score of the Translation suggestion in target language using language model and estimated length of actual translation of test sentence. We have used SRILM tool to generate 5 gram Bengali language model on the Bengali corpus used in Translation Memory. SRILM stores the probabilities in logarithmic form; hence they are negative in values. We have used N-Gram back-off model to estimate the language model probability. For every N-Gram match, we add the logarithmic score and get a negative score for that translation, let's say this score is -lm. Take the absolute value of this score and normalize it with length of Translation suggestion (TL), it becomes $P=lm/TL$. Then if we make the inverse of it($1/P$), we will get actual LM score($LMS=1/P$) for this translation.

We also used the concept of Brevity penalty to penalize a translation if its length is much smaller or much higher than the estimated reference translation. We don't have reference translation for the test sentence, but in our system we tried to estimate the length of reference translation based on the length of test sentence in source language. Let's assume length of test sentence is SL and length of Translation suggestion is TL. We have assumed that reference translation length will be in range of $0.8*SL$ to $1.2*SL$. If translation suggestion length falls in that range, then we don't give any penalty. But if the length of translation is out of that range we give a penalty based on following algorithm. We call this penalty as Length Based Penalty (LBP).

```
double GetLBP (int SL, int TL)
```

```

{
  int minRefLength=0.8 * SL;
  int maxRefLength=1.2 * SL;
  if (TL>=minRefLength AND TL<=maxRefLength)
  {
    LBP=1.0;
    return LBP;
  }
  else if(TL<minRefLength)
  {
    diff=minRefLength-TL;
  }
  else if(TL>maxRefLength)
  {
    diff=TL-maxRefLength;
  }
  LBP=e^(-diff/SL);
  return LBP;
}

```

We calculate smoothness score using language model score LMS and length based penalty BP.

$$\text{smoothness_score} = \text{LMS} * \text{LBP}$$

$$\text{Final_score} = \text{smoothness_score} * \text{NewTMMatchScore}$$

We re-rank the top suggestions based on this new score.

5.12 Experiments and Results

The effectiveness of the proposed system (CATaLog_MS) is demonstrated by comparing against CATaLog (Nayek et al., 2015 and the moses (Koehn et al., 2007)

implementation of the PB-SMT model. We have used a English to Bengali parallel corpus which has nearly 13000 sentences for training of our CATaLog and CATaLog_MS system. For building the PB-SMT system, we have used the same parallel corpus and we have used the maximum phrase length of 7 and a 5-gram language model trained using KenLM (Heafield, 2011). Parameter tuning was carried out using Minimum Error Rate Training (MERT) (Och, 2003) on the held-out development set.

We tested our system for English to Bengali translation. Two different test sets were used for evaluation: **Testset1** contained 100 sentences and **Testset2** contained 500 sentences. We evaluated our system using two well known automatic MT evaluation metrics: BLEU (Papineni et al., 2002) and TER (Snover et al., 2006).

CATaLog_MS provides five translation suggestions based on the top five close matches retrieved by CATaLog from the TM. The term 'First' in table Table1 represents the first (i.e. the top ranked) translation suggestion provided by CATaLog or CATaLog_MS system. The 'Best' translation suggestion among the five translation suggestions is chosen according to S-BLEU and TER. The purpose of using S-BLEU is to measure whether CATaLog can also be able to provide best suggestion or not.

Table1 shows that, as far as the 'First' translation suggestion is concerned, CATaLog_MS provides 2.13 and 2.03 BLEU points (22.4% and 19.2% relative) improvements over CATaLog for testset1 and testset2 respectively. The respective improvements are 8.21 and 9.64 points (12.8% and 14.6% relative) for TER.

Similarly, for the 'Best' translation suggestion, the improvements provided by CATaLog_MS over CATaLog for testset1 and testset2 are 3.59 and 1.91 BLEU

points (29.8% and 14.5% relative) and 10.99 and 6.24 TER points (17.1% and 10.3% relative) respectively.

More importantly, for testset1, CATaLog_MS `Best` performs better than the state-of-the-art PB-SMT system in both BLEU and TER. However, in case of testset2, CATaLog_MS `Best` performs better according to TER while Moses fares better according to BLEU. This is probably due to the fact that the Moses system was tuned with BLEU evaluation metric. The proposed system is performing better for both test set even if we consider only `First` translation suggestion.

From Table1, we can conclude that CATaLog_MS always performs better than CATaLog. TER score for CATaLog_MS is much lower than CATaLog for both `First` and `Best` translation suggestions. BLEU scores also reflect the same improvement over CATaLog. Comparison with Moses system reveals that CATaLog_MS provides lowest TER scores for both the test sets, even if we just consider the `first` translation suggestion. However, Moses is ahead on testset2 while CATaLog_MS fares better on testset1 according to BLEU.

Table 2 : Result of CATaLog System

Test Set	System		Performance	
			TER	BLEU
Test Set 1 (100 English Sentences)	CATaLog	First	0.6410	9.49
		Best	0.6041	12.03
	Moses		0.5712	14.57
	CATaLog_MS	First	0.5589	11.62
		Best	0.5342	15.62
	CATaLog_MS_ReRank		0.4849	18.07
Test Set 2 (500 English Sentences)	CATaLog	First	0.6598	10.58
		Best	0.6082	13.15
	Moses		0.5844	18.34
	CATaLog_MS	First	0.5634	12.61
		Best	0.5458	15.06
	CATaLog_MS_ReRank		0.5383	15.68

Above result shows that after re-ranking of top suggestions, if we just choose the first option, we are getting much higher BLEU score and lower TER score than the Moses in testset1. Though in testset2, BLEU score of CATaLog_MS_ReRank system with its first option is better than the best option of CATaLog_MS system, but lowers than that of Moses. However, for either case (i.e, testset1 and testset2), the TER score of CATaLog_MS_ReRank system is considerably better than the

other systems. It is to be noted that the CATaLog_MS `Best' system was decided on the basis of S-BLEU score, while for the actual evaluation purpose we use BLEU. BLEU being a system level score does not perform well at sentence level evaluation; hence the BLEU and TER scores provided by CaTaLog_MS_ReRank system are better than those provided by CATaLog_MS `Best' system.

Another important aspect to analyze the performance of a system is time complexity. We have observed that time complexity of our system mostly depends on size of translation memory. To compare a test sentence with large TM will increase the time complexity. Time taken to insert the meaning of unmatched word only for top 5 suggestions is quite insignificant with respect to TM comparison. This is reason we have used length based pruning of TM corpus, inverted index concept to reduce size of TM. These steps significantly reduce the TM comparison time.

Automatic Post Editing

Previously we have discussed how errors in machine translation output can be rectified by human post editors and can generate accurate translation. But this process involved human intervention. Question is can we automate this process? Is it possible to rectify the errors present in machine translation output automatically? This research field is called automatic post editing (APE). It is a very new research field in machine translation domain and very much challenging task. Real challenge here is that one has to model the behavior of a human post editor which is very difficult.

In automatic post editing, source language sentences, their machine translations and actual translations of source sentences by human translator are used for training the system. In testing phase, a source sentence and its machine translation is given as input to the APE system, and system has to predict the human generated translation without any human intervention. Machine translation system whose output is used as MT output can be any system. It should be considered just as a black box. In most of the cases, traces from the MT system are not available to use in APE system. But in some case, if it is available, it can be used to increase performance. We have tried to build APE system without using any trace from original machine translation system.

6.1 Related Research Work

Post processing of MT output has been a topic of research for long time. Aim of this research is to improve the quality of the MT output. In any case MT output has to be post edited by human post editors to produce publishable translations. But if post processing increase the quality of translation automatically, then post editing effort will be reduced significantly. It will save lot of money and time. Automatic post

editing of MT output can be seen as extension of that research. Here repetitive errors of MT system are rectified or human post editor's behavior is simulated to generate the actual translation.

Chen and Chen (1997) very first proposed an idea to combine RBMT and SMT system to produce a better translation. Combined system uses the some of the best features of both RBMT and SMT system to produce the desired result.

Dugast et al. (2007) uses statistical machine translation system on the output of rule based 'SYSTRAN' MT system output to automatically post edit it. They ran their experiment for English-French language pair and got 10 point increment in BLEU score, which is very significant.

Simard et al.(2007a) and Simard et al.(2007b) have shown that phrase based statistical machine translation system can give good result if used as automatic post editing system on rule based machine translation system output. Their experiment on English to French and French to English rule based machine translation system produced better BLEU and TER score. Not only that, their system's score was better than a standalone SMT system without any post editing system attached with it.

Some researchers used pre-processing of phrase table to produce better translation, than post process it. Eisele et al. (2008) combined multiple MT system to build a hybrid system. They combined the phrase translations generated from many rule based machine translation tools into the phrase table of a standard SMT tool Moses. They didn't change the decoding process of Moses. Moses chose the best phrase translation from that hybrid phrase table to generate the translation. They used this hybrid system for English-German, German-English, English-French, French-

English, English-Spanish and Spanish-English language pair on Europarl data and got 4 to 5 point BLEU score increment than standard Moses BLEU score.

Lagarda et al. (2009) used a statistical machine translation system as automatic post editing system and tried to rectify output of a rule based machine translation system. They compared the results of a RBMT system, SMT system and their APE system both using manual evaluation and automatic evaluation methods and showed that APE system performs better than other two systems.

Rosa et al. (2012) and Marecek et al. (2011) used hand written rules to rectify the frequent errors in system generated translations. They applied this system on a English-Czech MT system outputs and improves fluency of the translations and corrected some morphological errors too.

Parton et al. (2012) presented an innovative way to correct the MT outputs. They suggested three stages system to perform the APE task- detecting the errors, ranked suggestions of possible corrections and applying those suggestions. They proposed two methods to apply the suggestions- one is a rule based approach and another is feedback based approach. In rule based approach system directly applies the corrections and in feedback approach, system passed the possible correction to the MT decoder and decoder decides to apply it or not. Results shows improvement in translation in both rule based approach and feedback based approach.

Denkowski (2015) has developed method to learn from real time post editing of MT outputs and apply those adaptive rules to future system translation automatically to improve the system outputs. His experiments showed improvement in translation on English to Spanish and Spanish to English translation system.

Pal et al. (2015) used word alignment from multiple word aligners and combines them into single alignment. They used this hybrid alignment into a phrase based statistical machine translation system to improve the quality of the translation. They applied their system for English to Spanish machine translation and got better BLEU and TER score.

6.2 Dataset

We have used WMT 2016 automatic post editing task data for our experiments. It has 12000 English-German triplets for training. First part of the triplet is the source sentence in English. Second part of the triplet is MT output of the source sentence. This MT output has to be used like black box. Information about decoding process or corpus used in that system is not available. Third part of the triplet is human post edited translation of the MT output. Our APE system has been trained on these 12000 triplets. WMT 2016 provides another set of 1000 triplets which we have used to test our system. Source sentence part and MT output of these triplets are used as input to the APE system, and human post edited part of the triplet has been used as reference translation to evaluate the APE system automatically. We have used TER and BLEU score to evaluate our APE system.

6.3 Baseline APE System

We have used Moses SMT system as our baseline APE system. We have used MT output and human post edited output of training triplets as parallel corpus to train the translation model and used the human post edited output as monolingual corpus to build the language model. MT output part of the test triplets are given to this APE baseline system as input and decoded translation of this system is our final baseline APE output. Result of this experiment is shown in following table.

Table 3 : APE Baseline System Result

System Name	BLEU	TER
Original MT	60.09	24.42
Baseline APE	61.32	24.58

From above results we can see that though BLEU score has improved in baseline APE system, but TER value has been deteriorated.

6.4 Common Errors in SMT Output

Human post editors sometimes may generate the actual translation from scratch, rather than using the MT output. But in automatic post editing generating the translation from scratch without using the MT output means just building another MT system. So APE systems always use the MT output and from that, it tries to generate the actual translation. We can think of correcting the common mistakes appears in MT output in APE system to generate a better translation. Errors may happen in MT translation due to language model, translation model or may be due to decoding process. If we can correct some of those errors, we can get better translation than a MT system. Some of the very common errors found in MT outputs are follows:

1. Some words which should not be present in actual translation are present in system translation. If these words can be identified and then deleted from system translation, translation will become much closer to actual translation.

MT: Sie können auch **auf** Simplex- , Duotones , Triplex- und Quadruplexbilder in Photoshop **zu** erstellen .

Actual: Sie können auch Simplex- , Duplex- , Triplex- und Quadruplexbilder in Photoshop erstellen .

Red colored words in MT output are deleted in actual translation.

2. Some words which should appear in actual translation are not part of MT output. We need to insert these words in actual translation. Inserting words is much more difficult than deleting one.

MT: Beim Schließen eines Dokuments werden die Historie .

Actual: Beim Schließen eines Dokuments wird das **zugehörige Protokoll gelöscht** .

Red colored words are inserted in actual translation.

3. Some words in MT output should have been in different surface form. MT has chosen right root word for them, but inflected form it has used is not the right one. These inflectional errors can be corrected using APE system.

MT: Die Auto-Farbkorrekturoptionen können Sie Schatten- und Beschneidung **Prozentsätze** , und weisen Sie die Farbwerte für Schatten , Mitteltöne und Lichter angeben .

Actual: Mit den Auto-Farbkorrekturoptionen können Sie **Prozentwerte** für das Beschneiden von Tiefen und Lichtern festlegen sowie Tiefen , Mitteltönen und Lichtern Farbwerte zuweisen .

Here red colored word in MT and actual has same root word, but different surface form.

4. Position of some words in MT output may not be correct. APE system needs to reorder the words to generate a better translation.

MT: Illustrator weist **automatisch** eine gültige XML-ID , um alle dynamischen Objekte , die Sie erstellen .

Actual: Illustrator weist allen dynamischen Objekten zu , die Sie erstellen , **automatisch** eine gültige XML-ID .

Position of red colored word in MT and actual translation are quite different.

6.5 Our Approach

We have tried to fix the issue 1 and issue 3 describes in above section. We have tried to predict and delete those machine translation words in MT output which should not be present in actual translation. We have built two different MT word deletion model – one based source word to target word alignment and one based on source words context. We have used language model of target language to fix surface form error for some machine translated words.

6.5.1 Word Alignment Based MT Word Deletion Model

We have built a statistical model on the training triplets based on for a particular source word, how many times MT system translated that source word to a particular target word and how many times that target word got deleted from human post edited translation. Following is an example of this statistics:

Table 4 : MT Word Deletion Statistics Format

umzukehren		
Source_word	Total_frequency	Deletion_frequency
reverse	5	2
invert	1	1

It means English word ‘reverse’ translated to ‘umzukehren’ 5 times by the original MT system, out of which it has been deleted 2 times and word ‘invert’ translated to ‘umzukehren’ once by MT system and once it got deleted in actual translation.

Since we don’t have any information of MT decoding process and how human translation happened, we need to build this statistics by our own methods. We have

used GIZA++ alignment between source sentence and MT output and TER alignment between MT output and human post edited output to generate this statistics. GIZA++ alignment will give which source word translated to which target word in MT output and TER will give us if that target word got deleted in actual translation or not. From this information we can build this statistics which we called '**Deletion Probability Table**'. This table will contain the deletion probability for training triplets for source word and target word pair.

To test our system, we will give the source sentence and its MT output as input to our system. And we will delete those words from the MT output which will have high deletion probability in '**Deletion Probability Table**'. We can choose a cut off value say 0.5, if deletion probability of a MT word and aligned source word pair is higher than 0.5 we will delete that MT word, otherwise that MT word remain unchanged in APE output. But we don't have the alignment between test source sentence and its MT output. We will build this alignment from '**Deletion Probability Table**'. We will assume that every word of MT output can be aligned to every word of source sentence. If we merge pair wise source word and target word alignment information from '**Deletion Probability Table**', we can calculate the probability of a target word to be aligned with a particular source word. We calculate all the possible probabilities of alignments and sort them in descending order. We keep the maximum probable alignments as the desired one. After getting this alignment, we calculate the deletion probability of each target word and if those probabilities are higher than a pre-defined cut off probability, we delete that target word. We have run our experiments with following settings:

1. First, we have used the source sentence and original MT output along with deletion probability cut off values as 0.3, 0.5 and 0.7.
2. Second, we have used the source sentence and baseline APE system output with deletion probability cut off values 0.3, 0.5 and 0.7.

We have evaluated our system output based on BLEU and TER score and Precision-Recall value.

Precision = count of words correctly deleted by our system / count of words deleted from MT output by our system

Recall = count of words correctly deleted by our system / count of words actually deleted from MT output by human post editors

Our reference translation for test sentences are not tagged by human post editors, only actual translation is given. We have user TER alignment between MT output and human post edited output to calculate the number of words actually deleted from MT output by human post editors and number of words correctly deleted by our system.

Results of our experiment are shown in below table.

Table 5 : Result of APE Deletion Experiment

System Name	Deletion Probability Cut Off	Precision (P) and Recall (R)	BLEU	TER
With Original MT output	0.3	P=36.90,R=23.97	49.07	28.87
	0.5	P=43.55,R=11.65	55.89	25.72
	0.7	P=48.80,R=05.53	58.57	24.80
With Baseline APE output	0.3	P=30.71,R=18.91	51.22	28.62
	0.5	P=34.41,R=08.05	57.81	25.61
	0.7	P=35.51,R=03.23	60.07	24.92

Above result shows that if we can get high precision with a reasonable recall score, we can improve the TER score and BLEU score. For lower deletion probability cut off recall becomes high, but precision decreased which leads to high TER score and

low BLEU score. But if we increase the cut off value, we can get higher precision. Above result also shows that if we apply our system on output of baseline APE system rather than original MT output, we can get better TER and BLUE score for some test environment.

TER gives source to target sentence alignment based on the surface form of a word. Sometimes words in reference translation and machine translation output may appear in different surface form, but root words are same. TER can't detect this kind of matching. We have implemented detection of this kind of matching and removed those words from deletion statistics. This experiments leads to better result than the previous one. Following table shows the results, we got using stemming feature.

Table 6 : Result of APE Deletion Experiment with Word Stemming

System Name	Deletion Probability Cut Off	Precision (P) and Recall (R)	BLEU	TER
With Original MT output	0.3	P=39.45,R=18.28	52.32	27.23
	0.5	P=45.71,R=09.20	57.03	25.27
	0.7	P=50.00,R=04.47	58.85	24.69
With Baseline APE output	0.3	P=32.29,R=14.16	54.31	27.12
	0.5	P=36.33,R=06.47	58.77	25.25
	0.7	P=37.98,R=02.63	60.27	24.85

6.5.2 Source Word Context Based MT Word Deletion Model

‘Word Alignment based MT Word Deletion Model’ is highly dependent on bilingual alignment generated by GIZA++. But alignment generated by GIZA++ tool is not perfect. Errors in alignment can lead to wrong statistics. We took another approach to predict if a MT system translated word should be present or deleted in final translation or not. This approach is based on source words context. From the training triplets we can generate source words context vector for a particular target word in

MT translation for which this target word can be retained in final translation or can be deleted in final translation. TER alignment will give which target word in MT will be retained in final translation and which will be deleted. If a particular target word is deleted in final translation, then all the source words will be added to ‘**deletion source context vector**’ for that target word. Otherwise source words will be added to ‘**retention source context vector**’. From entire training triplets we will get two vectors for each target word presents in MT system translation – one for its deletion and other for its retention. A vector consists of source words along with their frequency. Vectors for deletion and retention are stored in following format.

Target Word: schärfer

Source Words: The 1 Sharpen 1 tool 1 sharpens 1 areas 1 in 1 an 1 image 1 . 1

For test MT sentence, for each target word we will calculate the cosine similarity between test source sentences and ‘**deletion source context vector**’ for that target word and another cosine similarity between test source sentence and ‘**retention source context vector**’ for that target word. Based on the cosine similarity score we can predict if that target word should be present in final translation or not.

Table 7 : Result of Source Word Context based MT Word Deletion Model

System Name	Precision (P) and Recall (R)	BLEU	TER
With Original MT output	P=27.70 R=34.35	38.33	35.65
With Baseline APE output	P=24.39 R=30.93	40.26	35.46

6.5.3 Word Deletion Model

Table 4 gives us an idea that if we use higher deletion probability cut off, we can achieve better precision, but it reduces recall. Table 5 shows that source words context based deletion prediction can give higher recall, but precision is poor here. We can combine ‘deletion probability table’ statistics with this ‘source context vector statistics’ to test if this combination can give satisfactory precision and recall. In this combined system we have deleted those target words in final translation for which ‘deletion probability’ is more than cut off probability and its cosine score for deletion source context vector is more than the cosine score of retention source context vector. This final is called ‘Word Deletion Model’. Following table shows our experiments result.

Table 8 : Result of Combination of Word Deletion Model

System Name	Deletion Probability Cut Off	Precision (P) and Recall (R)	BLEU	TER
With Original MT output	0.3	P=46.78 R=12.31	56.20	25.47
	0.5	P=54.06 R=05.84	58.76	24.59
	0.7	P=53.47 R=03.72	59.25	24.53
With Baseline APE output	0.3	P=37.68 R=09.17	57.92	25.48
	0.5	P=40.39 R=03.75	60.19	24.74
	0.7	P=36.17 R=02.05	60.60	24.71

6.5.4 Surface Form Correction Model

Another issue, we have tried to fix is the issue 3 mentioned in section 6.4. Fixing all the MT words which appears in wrong morphological form in MT output is difficult.

The search space for fixing all morphological errors is huge and it will take lot of time to fix one single sentence. So we have tried to fix just a single target word which should be present final translation but in another surface form. We have used German monolingual corpus to generate all possible surface form of a particular root word using StemmersNet²⁰. Following is an example of these surface forms is stored.

Root word: europa

Surface Forms: europäische | europäischen | Europäischen | Europäische | Europa
Europäer | europäischer | europäisches | Europäern | Europäisches | europäisch |
europäischem

We have used SRILM tool to calculate language model probability to choose the better translation. We assumed that in a particular MT output there can be maximum one target word whose surface form has to be changed in final translation. So for each target root word we will try all its surface form one by one and calculate the probability of that sentence. We will choose that sentence which will get highest language model probability. We will consider the original MT output also in this process.

Table 9 : Result of Surface Form Correction Model

System Name	BLEU	TER
With Original MT output	60.01	24.46
With Baseline APE output	61.49	24.50

²⁰ <https://stemmersnet.codeplex.com/>

6.5.5 Combination of Word Deletion Model and Surface Form Correction Model

Next experiments I have done it to combine the surface correction feature with the deletion prediction feature. First using deletion prediction feature, deleted those target words which have high deletion chances. Then I have applied the surface correction on other target words. Result of this experiments in shown in following table.

Table 10 : Result of Word Deletion and Surface Form Correction Model

System Name	Deletion Probability Cut Off	BLEU	TER
With Original MT output	0.3	56.00	25.61
	0.5	58.62	24.67
	0.7	59.10	24.61
With Baseline APE output	0.3	57.97	25.49
	0.5	60.29	24.69
	0.7	60.70	24.67

6.5.6 Automatic Post Editing Model

From all the analysis mentioned above, I have noticed that if we keep deletion probability cut off value at 0.5 we are getting good precision and recall. So we fixed it at that value. Now we have total 8 hypothesis translation file for the original source file. These translations are following:

1. Original MT output – called DE_{MT}
2. Surface Form Correction Model applied on DE_{MT} – called $DE_{SC_{MT}}$
3. Word Deletion Model applied on DE_{MT} – called $DE_{D_{MT}}$
4. Word Deletion Model and Surface Correction Model applied on DE_{MT} – called $DE_{D_{SC_{MT}}}$
5. Baseline APE output – called DE_{BAPE}
6. Surface Form Correction Model applied on DE_{BAPE} – called $DE_{SC_{BAPE}}$

7. Word Deletion Model applied on DE_{BAPE} – called DE_D_{BAPE}
8. Word Deletion Model and Surface Form Correction Model applied on DE_{BAPE} – called $DE_D_SC_{BAPE}$

I have used ranking algorithm which is used in ‘CATaLog with MS Fusion’ system to rank above 8 hypothesis translation for every source sentence and generated best translation file. Following is the result of analysis of ranked system.

Table 11 : Result of APE Model

System Name	BLEU	TER
Original MT	60.09	24.42
Baseline APE	61.32	24.58
Ranked System	61.94	24.02

We can see that BLEU score has increased for ranked system than the original MT system and baseline system and TER also got reduced than both the system. Our experiment shows that improvement can be made on machine translation output by rectifying common errors which occurred while statistical machine translation decoding. But more research needed to improve the translation quality significantly.

Conclusions

CATaLog tool is specifically targeted towards improving user experience with TM. It does so by color coding the TM suggestions. Color coding helps human post editors to choose the best option among many others. In CATaLog_MS, we have introduced another important functionality in TM, that of proposing a new translation. Traditionally, TMs do not generate any translation; so we present a step beyond traditional TM. Besides, this improves HCI issues with TM since this new functionality generates a new translation based on the translation template chosen by the user.

Automatic Post Editing can be used to replace the human post editors. It is very difficult to model human post editor's behavior. That's why research in APE is in very early stage and not much progress has been made. We have tried to fix some common errors in MT output, but more exhaustive research needed to generate publishable APE output.

7.1 Scope of future work

In our current work if an isolated word matches between input sentence and reference source sentence that word is assumed as a match. But in practical scenario such isolated word may not be a proper match. We can build a classifier which can tell if an isolated word is actually a match or not. We can use the context of the word, its POS tag; parse tree features to build this classifier. We can then include this feature in similarity score calculation technique and can produce much better translation options for post editors.

Punctuation like comma can play a major role in deciding which reference sentences are closer. Rather than just considering matches with entire sentence, we can try to

match punctuated portions of sentences. A phrase match appearing within punctuation should be given more weight than a match which is mingled between two punctuated phrases.

We can use named entity tagging, named entity list and gazetteer to translate named entities. This will reduce translation time for post editors. Names entity tagger can be use to tag named entities in test sentences. If any selected sentence has a different named entity, we can replace that with actual named entity of test sentence. Replacing named entities will reduce significant amount of post editing time.

In future we will replace the existing bilingual dictionary with a probabilistic bilingual dictionary.

We would like to conduct a user evaluation in real world experimental settings with human translators to measure productivity gain yielded by the tool. We would also like carry out an evaluation to compare our system against other CAT systems available.

For APE system, we need to build a model to insert new words which should appear in actual translation. If we can insert new words in MT output we can achieve better BLEU or TER score. But building this model needs large amount of training corpus. Availability of large training corpus is a major bottleneck for APE research. Regarding surface correction model we are just trying to correct one word. Correcting all the morphological errors will increase the search space and it will take lot of time to generate a single APE output. Some kind of sub optimal searching algorithm can be used to rectify as many as morphological errors which will lead to better result.

Bibliography

1. Tapas Nayek, Sudip Kumar Naskar, Santanu Pal, Marcos Zampieri, Mihaela Vela, Josef van Genabith (2015) CATaLog: New Approaches to TM and Post Editing Interfaces. Proceedings of the Workshop on Natural Language Processing for Translation Memories (NLP4TM). Hissar, Bulgaria.
2. Tapas Nayak, Santanu Pal, Sudip Kumar Naskar, Sivaji Bandyopadhyay and Josef van Genabith. 2016. Beyond Translation Memories: Generating Translation Suggestions based on Parsing and POS Tagging. To be published in the Proceedings of the 2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM 2016), Portorož, Slovenia.
3. Yifan He, Yanjun Ma, Josef van Genabith, & Andy Way: Bridging SMT and TM with translation recommendation. ACL 2010: the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, July 11-16, 2010: Conference proceedings; pp.622-630. [PDF, 573KB]
4. Panagiotis Kanavos & Dimitrios Kartsaklis: Integrating machine translation with translation memory: a practical approach. JEC 2010: Second joint EM+/CNGL Workshop “Bringing MT to the user: research on integrating MT in the translation industry”, AMTA 2010, Denver, Colorado, November 4, 2010; pp.11-20. [PDF, 618KB]
5. Philipp Koehn & Jean Senellart: Convergence of translation memory and statistical machine translation. JEC 2010: Second joint EM+/CNGL Workshop “Bringing MT to the user: research on integrating MT in the translation industry”, AMTA 2010, Denver, Colorado, November 4, 2010; pp.21-31. [PDF, 188KB]

6. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In Proc. of the 45th Annual meeting of the Association for Computational Linguistics (ACL 2007), pp. 177- 180.
7. Sandipan Dandapat, Sara Morrissey, Abdy Way, & Joseph van Genabith: Combining EBMT, SMT, TM and IR technologies for quality and scale. EACL Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra): Proceedings of the workshop, 23-24 April 2012, Avignon, France; pp.48-58. [PDF, 248KB]
8. Tapas Nayek, Sudip Kumar Naskar, Santanu Pal, Marcos Zampieri, Mihaela Vela and Josef van Genabith. CATaLog:New Approaches to TM and Post Editing Interfaces. In the Proceedings of the 1st Workshop on Natural Language Processing for Translation Memories (NLP4TM), Hissar, Bulgaria.
9. Bici, E. and Dymetman, M. (2008). Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. Computational Linguistics and Intelligent TextProcessing, pages 454–465.
10. Dandapat, S., Morrissey, S., Way, A., and Forcada, M. L. (2011). Using example-based MT to support statistical MT when translating homogeneous data in a resource-poor setting. In Proceedings of the 15th conference of the European Association for Machine Translation, pages 201–208. Leuven, Belgium.
11. Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with

- a Tunable MT Metric. In Proceedings of the Fourth Workshop on Statistical Machine Translation, EACL 2009.
12. Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
 13. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, & Robert L. Mercer: The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19 (2), pp. 263-311.
 14. Philipp Koehn, Franz Josef Och and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pages 48–54.
 15. Simard, M. and Isabelle, P. (2009). Phrase-based machine translation in a computer-assisted translation environment. *Proceeding of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120–127.
 16. Zhechev, V. and Genabith, J. V. (2010). Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of SSST-4 - 4th Workshop on Syntax and Structure in Statistical Translation*, pages 43–49, Dublin, Ireland.
 17. Ma, Y., He, Y., Way, A., and van Genabith, J. (2011). Consistent translation using discriminative learning - a translation memory-inspired approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1239–1248, Portland, Oregon, USA. Association for Computational Linguistics.
 18. Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904.

19. Kenneth Heafield. WMT at EMNLP, Edinburgh, Scotland, United Kingdom, 30—31 July, 2011.
20. Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In Proceedings of Interspeech, Brisbane, Australia.
21. Adam Pauls and Dan Klein. 2011. Faster and smaller ngram language models. In Proceedings of ACL, Portland, Oregon.
22. David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In Proceedings of ACL, pages 512–519, Prague, Czech Republic.
23. Kuang-Hua Chen and Hsin-Hsi Chen. 1997. A Hybrid Approach to Machine Translation System Design. *Computational Linguistics and Language Processing*, 23:241–265.
24. Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies (NAACL-HLT), pages 427–436.
25. Donald De Palma and Nataly Kelly. 2009. Project Management for Crowd sourced Translation: How User-Translated Content Projects Work in Real Life. *Translation and Localization Project Management: The Art of the Possible*, pages 379–408.
26. Michael Denkowski. 2015. Machine Translation for Human Translators. Ph.D. thesis, Carnegie Mellon University. Rebecca Fiederer and Sharon O’Brien. 2009. Quality and Machine Translation: a Realistic Objective. *Journal of Specialised Translation*, 11:52–74.

27. George Foster, Roland Kuhn, and Howard Johnson, 2006. Phrase table Smoothing for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 53–61.
28. Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 848–856.
29. Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In Proceedings of the 6th Workshop on Statistical Machine Translation (WMT), pages 187–197.
30. Marcin Junczys-Dowmunt and Arkadiusz Szał. 2012. SyMGiza++: Symmetrized Word Alignment Models for Statistical Machine Translation. In Proceedings of the International Conference on Security and Intelligent Information Systems (SIIS), pages 379–390.
31. Kevin Knight and Ishwar Chander. 1994. Automated Post-Editing of Documents. In Proceedings of the 12th National Conference on Artificial Intelligence, pages 779–784.
32. Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT), pages 48–54, Stroudsburg, PA, USA.
33. Philipp Koehn. 2009. A Process Study of Computer Aided Translation. *Machine Translation*, 23(4):241–263.
34. Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

35. Antonio Lagarda, Vicent Alabau, Francisco Casacuberta, Roberto Silva, and Enrique D'íaz-de Liaño. 2009. Statistical Post-Editing of a Rule-based Machine Translation System. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies (NAACL-HLT), pages 217–222 Stroudsburg, PA, USA.
36. Alon Lavie and Abhaya Agarwal. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the 2nd Workshop on Statistical Machine Translation (WMT), pages 228–231.
37. Santanu Pal, Mihaela Vela, Sudip Kumar Naskar, Josef van Genabith. 2015. USAAR-SAPE: An English–Spanish Statistical Automatic Post-Editing System. In the Proceedings of the EMNLP 2015 Tenth Workshop on Statistical Machine Translation (WMT 2015), Lisbon, Portugal.
38. Loic Dugast, Jean Senellart and Philipp Koehn, 2007. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In the Proceedings of the Second Workshop on Statistical Machine Translation, pages 220–223, Prague, June 2007.
39. Michel Simard, Cyril Goutte and Pierre Isabelle, 2007. Statistical Phrase Based Post Editing. In the Proceedings of NAACL-HLT, Canada.
40. Michel Simard, Nicola Ueffing, Pierre Isabelle and Roland Kuhn. Rule-based Translation With Statistical Phrase-based Post-editing. In the ACL 2007 Second Workshop on Statistical Machine Translation. Prague, Czech Republic. June 23, 2007.
41. Andreas Eisele, Christian Federmann, Hervé Saint-Amand, Michael Jellinghaus, Teresa Herrmann, Yu Chen. Using Moses to Integrate Multiple Rule-Based Machine Translation Engines into a Hybrid System. In the

- Proceedings of the Third Workshop on Statistical Machine Translation, pages 179–182, Columbus, Ohio, USA, June 2008.
42. Miquel Esplà, Felipe Sánchez-Martínez, & Mikel L. Forcada: Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited. In the Proceedings of the 15th conference of the European Association for Machine Translation, May 2011, Leuven, Belgium.
 43. Kun Wang, Chengqing Zong, and Key-Yih Su. Integrating translation memory into phrase-based machine translation during decoding. In the Proceedings of the 51st Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, August 2013.
 44. Satyanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan, June.
 45. Rudolf Rosa , David Mareček , Ondřej Dušek. DEPFIX: a system for automatic correction of Czech MT outputs, Proceedings of the Seventh Workshop on Statistical Machine Translation, June 07-08, 2012, Montreal, Canada.
 46. David Mareček , Rudolf Rosa , Petra Galuščáková, Ondřej Bojar. Two-step translation with grammatical post-processing, Proceedings of the Sixth Workshop on Statistical Machine Translation, July 30-31, 2011, Edinburgh, Scotland.