CATaLog: New Approaches to TM and Post Editing Interfaces

Tapas Nayek[1], Sudip Kumar Naskar[1], Santanu Pal[2], Marcos Zampieri[2,3], Mihaela Vela[2], Josef van Genabith[2,3]
Jadavpur University, India[1]
Saarland University, Germany[2]
German Research Center for Artificial Intelligence (DFKI), Germany[3]
tnk02.05@gmail.com, sudip.naskar@cse.jdvu.ac.in, m.vela@mx.uni-saarland.de
santanu.pal|marcos.zampieri|josef.vangenabith@uni-saarland.de

Abstract

This paper explores a new TM-based CAT tool entitled *CATaLog*. New features have been integrated into the tool which aim to improve post-editing both in terms of performance and productivity. One of the new features of *CATaLog* is a color coding scheme that is based on the similarity between a particular input sentence and the segments retrieved from the TM. This color coding scheme will help translators to identify which part of the sentence is most likely to require post-editing thus demanding minimal effort and increasing productivity. We demonstrate the tool's functionalities using an English - Bengali dataset.

1 Introduction

The use of translation and text processing software is an important part of the translation workflow. Terminology and computer-aided translation tools (CAT) are among the most widely used software that professional translators use on a regular basis to increase their productivity and also improve consistency in translation.

The core component of the vast majority of CAT tools are translation memories (TM). TMs work under the assumption that previously translated segments can serve as good models for new translations, specially when translating technical or domain specific texts. Translators input new texts into the CAT tool and these texts are divided into shorter segments. The TM engine then checks whether there are segments in the memory which are as similar as possible to those from the input text. Every time the software finds a similar segment in the memory, the tool shows it to the translator as a suitable suggestion usually through a graphical interface. In this scenario, translators work as post-editors by correcting retrieved segments suggested by the CAT tool or translating new segments from scratch. This process is done iteratively and every new translation increases the size of the translation memory making it both more useful and more helpful to future translations.

Although in the first place it might sound very simplistic, the process of matching source and target segments, and retrieving translated segments from the TM is far from trivial. To improve the retrieval engines, researchers have been working on different ways of incorporating semantic knowledge, such as paraphrasing (Utiyama et al., 2011; Gupta and Orăsan, 2014; Gupta et al., 2015), as well as syntax (Clark, 2002; Gotti et al., 2005) in this process. Another recent direction that research in CAT tools is taking is the integration of both TM and machine translation (MT) output (He et al., 2010; Kanavos and Kartsaklis, 2010). With the improvement of state-of-the-art MT systems, MT output is no longer used just for *gisting*, it is now being used in real-world translation projects. Taking advantage of these improvements, CAT tools such as MateCat[1], have been integrating MT output along TMs in the list of suitable suggestions (Cettolo et al., 2013).

In this paper we are concerned both with retrieval and with the post-editing interface of TMs. We present a new CAT tool called *CATaLog*[2], which is language pair independent and allows users to upload their own memories in

[1] www.matecat.com

[2] The tool will be released as a freeware open-source software. For more information, use the following URL: http://ttg.uni-saarland.de/software/catalog

the tool. Examples showing the basic functionalities of *CATaLog* are presented using English - Bengali data.

2 Related Work

CAT tools have become very popular in the translation and localization industries in the last two decades. They are used by many language service providers, freelance translators to improve translation quality and to increase translator's productivity (Lagoudaki, 2008). Although the work presented in this paper focuses on TM, it should also be noted that there were many studies on MT post-editing published in the last few years (Specia, 2011; Green et al., 2013; Green, 2014) and as mentioned in the last section, one of the recent trends is the development of hybrid systems that are able to combine MT with TM output. Therefore work on MT post-editing presents significant overlap with state-of-the-art CAT tools and to what we propose in this paper.

Substantial work have also been carried out on improving translation recommendation systems which recommends post-editors either to use TM output or MT output (He et al., 2010). To achieve good performance with this kind of systems, researchers typically train a binary classifier (e.g., Support Vector Machines) to decide which output (TM or MT) is most suitable to use for post-editing. Work on integrating MT with TM has also been done to make TM output more suitable for post-editing diminishing translators' effort (Kanavos and Kartsaklis, 2010). Another study presented a *Dynamic Translation Memory* which identifies the longest common subsequence in the the closest matching source segment, identifies the corresponding subsequence in its translation, and dynamically adds this source-target phrase pair to the phrase table of a phrase-based ststistical MT (PB-SMT) system (Biçici and Dymetman, 2008).

Simard and Isabelle (2009) reported a work on integration of PB-SMT with TM technology in a CAT environment in which the PB-SMT system exploits the most similar matches by making use of TM-based feature functions. Koehn and Senellart (2010) reported another MT-TM integration strategy where TM is used to retrieve matching source segments and mismatched portions are translated by an SMT system to fill in the gaps.

Even though this paper describes work in progress, our aim is to develop a tool that is as intuitive as possible for end users and this should have direct impact on translators' performance and productivity. In the recent years, several productive studies were also carried out measuring different aspects of the translation process such as cognitive load, effort, time, quality as well as other criteria (Bowker, 2005; O'Brien, 2006; Guerberof, 2009; Plitt and Masselot, 2010; Federico et al., 2012; Guerberof, 2012; Zampieri and Vela, 2014). User studies were taken into account when developing CATaLog as our main motivation is to improve the translation workflow. In this paper, however, we do not yet explore the impact of our tool in the translation process, because the functionalities required for this kind of study are currently under development in CATaLog. Future work aims to investigate the impact of the new features we are proposing on the translator's work.

3 System Description

We demonstrate the functionalities and features of *CATaLog* in an English - Bengali translation task. The TM database consists of English sentences taken from BTEC[3] (Basic Travel Expression Corpus) corpus and their Bengali translations[4]. Unseen input or test segments are provided to the post-editing tool and the tool matches each of the input segments to the most similar segments contained in the TM. TM segments are then ranked according their the similarity to the test sentence using the popular Translation Error Rate (TER) metric (Snover et al., 2009). The top 5 most similar segments are chosen and presented to the translator ordered by their similarity.

One very important aspect of computing similarity is alignment. Each test (input) segment in the source language (SL) is aligned with the reference SL sentences in the TM and each SL sentence in the TM is aligned to its respective translation. From these two sets

[3] BTEC corpus contains tourism-related sentences similar to those that are usually found in phrase books for tourists going abroad

[4] Work in progress.

of alignments we apply a method to find out which parts of the translation are relevant with respect to the test sentence and which are not, i.e., which parts of the TM translation should remain intact after post editing and which portions should be edited. After this process, matched parts and unmatched parts are color-coded for better visualization; matched parts are displayed in green and unmatched parts are displayed in red. The colors help translators to visualize instantaneously how similar are the five suggested segments to the input segment and which one of them requires the least effort to be post-edited.

3.1 Finding Similarity

For finding out the similar and dissimilar parts between the test segment and a matching TM segment, we use TER alignments. TER is an error metric and it gives an edit ratio (often referred to as edit rate or error rate) in terms of how much editing is required to convert a sentence into another with respect to the length of the first sentence. Allowable edit operations include *insert*, *delete*, *substitute* and *shift*. We use the TER metric (using tercom-7.251[5]) to find the edit rate between a test sentence and the TM reference sentences.

Simard and Fujita (2012) first proposed the use of MT evaluation metrics as similarity functions in implementing TM functionality. They experimented with several MT evaluation metrics, viz. BLEU, NIST, Meteor and TER, and studied their behaviors on TM performance. In the TM tool presented here we use TER as the similarity metric as it is very fast and lightweight and it directly mimics the human post-editing effort. Moreover, the tercom-7.251 package also produces the alignments between the sentence pair from which it is very easy to identify which portions in the matching segment match with the input sentence and which portions need to be worked on. Given below are an input sentence, a TM match and the TER alignment between them where C represents a match (shown as the vertical bar ‘|’), and I, D and S represents the three post-editing actions - insertion, deletion and substitution, respectively.

Input: we would like a table by the window .

TM Match: we want to have a table near the window .

TER alignment:

“we”,“we”,C,0
“want”,“”,D,0
“to”,“would”,S,0
“have”,“like”,S,0
“a”,“a”,C,0
“table”,“table”,C,0
“near”,“by”,S,0
“the”,“the”,C,0
“window”,“window”,C,0
“.”,“.”,C,0

```
we want  to    have a table near the window .
|    D    S     S   |   |    S    |     |    |
we   -   would like a table  by  the window .
```

Since we want to rank reference sentences based on their similarity with the test sentence, we use the TER score in an inverse way. TER being an error metric, the TER score is proportional to how dissimilar two sentences are; i.e., the lower the TER score, the higher the similarity. We can directly use the TER score for ranking of sentences. However, in our present system we have used our own scoring mechanism based on the alignments provided by TER. TER gives equal weight to each edit operation, i.e., deletion, insertion, substitution and shift. However, in post-editing, deletion takes much lesser time compared to the other editing operations. Different costs for different edit operations should yield in better results. These edit costs or weights can be adjusted to get better output from TM. In the present system, we assigned a very low weight to delete operations and equal weights to the other three edit operations. To illustrate why different editing costs matter, let us consider the example below.

- **Test segment:** how much does it cost ?
- **TM segment 1:** how much does it cost to the holiday inn by taxi ?
- **TM segment 2:** how much ?

If each edit operation is assigned an equal weight, according to TER score, TM segment

[5] http://www.cs.umd.edu/ snover/tercom/

2 would be a better match with respect to the test segment, as TM segment 2 involves inserting translations for 3 non-matching words in the test segment ("does it cost"), as opposed to deleting translations for 6 non-matching words ("to the holiday inn by taxi") in case of TM segment 1. However, deletion of translations for the 6 non-matching words from the translation of TM segment 1, which are already highlighted red by the TM, takes much less cognitive effort and time than inserting translations of 3 non-matching words into the translation of TM segment 1 in this case. This justifies assigning minimal weights to the deletion operation which prefers TM segment 1 over TM segment 2 for the test segment shown above.

3.2 Color Coding

Among the top 5 choices, post-editor selects one reference translation to do the post-editing task. To make that decision process easy, we color code the matched parts and unmatched parts in each reference translation. Green portion implies that they are matched fragments and red portion implies a mismatch.

The alignments between the TM source sentences and their corresponding translations are generated using GIZA++ (Och and Ney, 2003) in the present work. However, any other word aligner, e.g., Berkley Aligner (Liang et al., 2006), could be used to produce this alignment. The alignment between the matched source segment and the corresponding translation, together with the TER alignment between the input sentence and the matched source segment, are used to generate the aforementioned color coding between selected source and target sentences. The GIZA++ alignment file is directly fed into the present TM tool. Given below is an example TM sentence pair along with the corresponding word alignment input to the TM.

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

- **Bengali:** আমরা জানালার কাছে একটা টেবিল চাই ।

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

GIZA++ generates the alignment between TM source sentences and target sentences. This alignment file is generated offline, only once, on the TM database. TER gives us the alignments between a test sentence and the corresponding top 5 matching sentences. Using these two sets of alignments we color the matched fragments in green and the unmatched fragments in red of the selected source sentences and their corresponding translations.

Color coding the TM source sentences makes explicit which portions of matching TM source sentences match with the test sentence and which ones not. Similarly, color coding the TM target sentences serves two purposes. Firstly, it makes the decision process easier for the translators as to which TM match to choose and work on depending on the color code ratio. Secondly, it guides the translators as to which fragments to post-edit. The reason behind color coding both the TM source and target segments is that a longer (matched or non-matched) source fragment might correspond to a shorter source fragment, or vice versa, due to language divergence. A reference translation which 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 might have to insert translations for more non-matched words in the input sentence. In this context, it is to be noted that insertion and substitution are the most costly operations in post-editing. However, such sentences will not be preferred by the TM as we assign a higher cost for insertion than deletion, and hence such sentences will not be shown as the top candidates by the TM. Figure 1 presents a snapshot of CATaLog.

Input: you gave me wrong number .

Source Matches:

1. you gave me the wrong change . i paid eighty dollars .

2. i think you 've got the wrong number .

3. you are wrong .

4. you pay me .

5. you ’re overcharging me .

Target Matches:

1. আপনি আমাকে ভুল খুচরো দিয়েছেন . আমি আশি ডলার দিয়েছি . (*Gloss: apni amake vul khuchro diyechen . ami ashi dollar diyechi .*)
2. আমার ধারণা আপনি ভুল নম্বরে ফোন করেছেন . (*Gloss: amar dharona apni vul nombore phon korechen .*)
3. আপনি ভুল . (*Gloss: apni vul .*)
4. আপনি আমাকে টাকা দিন . (*Gloss: apni amake taka din .*)
5. আপনি আমার কাছে থেকে বেশি নিচ্ছেন . (*Gloss: apni amar kache theke beshi nichchen .*)

For the input sentence shown above, the TM system shows the above mentioned color coded 5 topmost TM matches in order of their relevance with respect to the post-editing effort (as deemed by the TM) for producing the translation for the input sentence.

3.3 Improving Search Efficiency

Comparing every input sentence against all the TM source segments makes the TM very slow. In practical scenario, in order to get good results from a TM, the TM database should be as large as possible. In that case determining the TER alignments will take a lot time for all the reference sentences (i.e., source TM segments). For improving the search efficiency, we make use of the concept of posting lists which is a de facto standard in information retrieval using inverted index.

We create a (source) vocabulary list on the training TM data after removing stop words and other tokens which occur very frequently and have less importance in determining similarity. All the words are then lowercased. Unlike in information retrieval, we do not perform any stemming of the words as we want to store the words in their surface form so that if they appear in the same form as in some input sentence, only then we will consider it as a match. For each word in the vocabulary we maintain a posting list of sentences which contain that word.

We only consider those TM source sentences for similarity measurement which contain one or more vocabulary word(s) of the input sentence. This reduces the search space and the time taken to produce the TM output. The CATaLog tool provides an option whether to use these postings lists or not. This feature is there to compare results using and without using postings lists. In ideal scenario, TM output for both should be the same, though time taken to produce the output will be significantly different.

3.4 Batch Translation

The tool also provides an option for translating sentences in bulk mode. Post-editors can generate TM output for an entire input file at a time using this option. In this case the TM output is saved in a log file which the post-editors can directly work with later in offline mode, i.e., without using the TM tool.

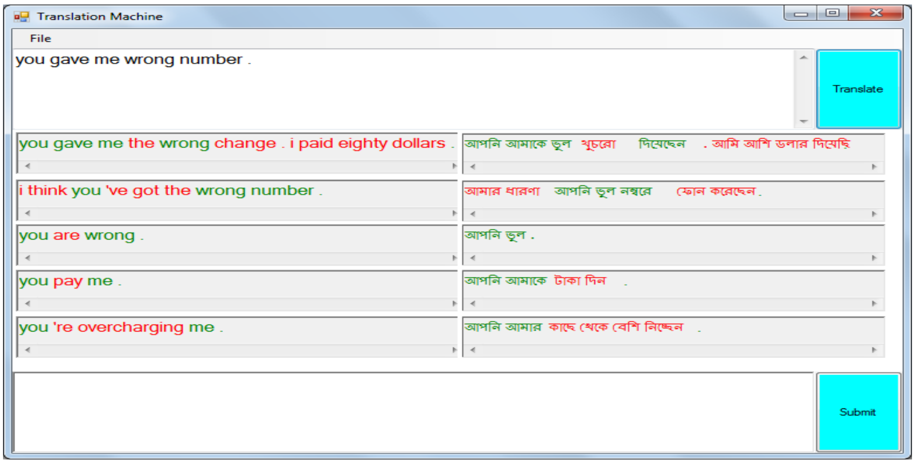
4 Conclusions and Future Work

This paper presents ongoing research and development of a new TM-based CAT tool and post-editing interface entitled *CATaLog*. Even though it describes work in progress, we believe some interesting new insights are discussed and presented in this paper. The tool will be made available in the upcoming months as an open-source free software.

We are currently working on different features to measure time and cognitive load in the post-editing process. The popular keystroke logging is among them. We would like to investigate the impact of the innovations presented here in real world experimental settings with human translators.

We are integrating and refining a couple of features in the tool as for example sentence and clause segmentation using comma and semi-colon as good indicators. It should also be noted that in this paper we considered only word alignments. In the future we would also like to explore how multi-word expressions (MWE) and named entities (NE) can help in TM retrieval and post editing.

We are also exploring contextual, syntactic and semantic features which can be included in similarity scores calculation to retrieve more appropriate translations. Another improvement we are currently working on concerns weight assignment to different edit operations.



you gave me wrong number .

you gave me the wrong change . i paid eighty dollars .	আপনি আমাকে ভুল খুচরো দিয়েছেন . আমি আশি ডলার দিয়েছি
i think you 've got the wrong number .	আমার ধারণা আপনি ভুল নম্বরে ফোন করেছেন .
you are wrong .	আপনি ভুল .
you pay me .	আপনি আমাকে টাকা দিন .
you 're overcharging me .	আপনি আমার কাছে থেকে বেশি নিচ্ছেন .

Figure 1: Screenshot with color cording scheme.

We believe these weights can be used to optimize system performance.

Finally, another feature that we are investing is named entity tagging. Named entity lists and gazetteers can be used to identify and to translate named entities in the input text. This will help reduce the translation time and effort for post-editors. The last two improvements we mentioned are, of course, language dependent.

Acknowledgements

We would like to thank the anonymous NLP4TM reviewers who provided us valuable feedback to improve this paper as well as new ideas for future work.

English to Indian language Machine Translation (EILMT) is a project funded by the Department of Information and Technology (DIT), Government of India.

Santanu Pal is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement no 317471.

References

Ergun Biçici and Marc Dymetman. 2008. Dynamic translation memory: using statistical machine translation to improve translation memory fuzzy matches. In *Computational Linguistics and Intelligent Text Processing*, pages 454–465. Springer.

Lynne Bowker. 2005. Productivity vs Quality? A pilot study on the impact of translation memory systems. *Localisation Reader*, pages 133–140.

Mauro Cettolo, Christophe Servan, Nicola Bertoldi, Marcello Federico, Loic Barrault, and Holger Schwenk. 2013. Issues in incremental adaptation of statistical MT from human post-edits. In *Proceedings of the Workshop on Post-editing Technology and Practice (WPTP-2)*, Nice, France.

J.P. Clark. 2002. System, method, and product for dynamically aligning translations in a translation-memory system, February 5. US Patent 6,345,244.

Marcello Federico, Alessandro Cattelan, and Marco Trombetti. 2012. Measuring User Productivity in Machine Translation Enhanced Computer Assisted Translation. In *Proceedings of Conference of the Association for Machine Translation in the Americas (AMTA)*.

Fabrizio Gotti, Philippe Langlais, Elliott Macklovitch, Benoit Robichaud Didier Bourigault, and Claude Coulombe. 2005. 3GTM: A Third-Generation Translation Memory. In *3rd Computational Linguistics in the North-East (CLiNE) Workshop*, Gatineau, Québec, aug.

Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Spence Green. 2014. *Mixed-initiative natural language translation*. Ph.D. thesis, Stanford University.

Ana Guerberof. 2009. Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus*, 7(1):133–140.

Ana Guerberof. 2012. *Productivity and Quality in the Post-Edition of Outputs from Translation Memories and Machine Translation*. Ph.D. thesis, Rovira and Virgili University Tarragona.

Rohit Gupta and Constantin Orăsan. 2014. Incorporating Paraphrasing in Translation Memory Matching and Retrieval. In *Proceedings of EAMT*.

Rohit Gupta, Constantin Orăsan, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. 2015. Can Translation Memories afford not to use paraphrasing? In *Proceedings of EAMT*.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with translation recommendation. In *Proceedings of ACL*, pages 622–630.

Panagiotis Kanavos and Dimitrios Kartsaklis. 2010. Integrating Machine Translation with Translation Memory: A Practical Approach. In *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*.

Philipp Koehn and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Elina Lagoudaki. 2008. The value of machine translation for the professional translator. In *Proceedings of the 8th Conference of the Association for Machine Translation in the Americas*, pages 262–269, Waikiki, Hawaii.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111.

Sharon O'Brien. 2006. Eye-tracking and translation memory matches. *Perspectives: Studies in Translatology*, 14:185–204.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Mirko Plitt and François Masselot. 2010. A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, 93:7–16.

Michel Simard and Atsushi Fujita. 2012. A Poor Man's Translation Memory Using Machine Translation Evaluation Metrics. In *Proceedings of the 10th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, San Diego, California, USA.

Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120–127.

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.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80, Leuven, Belgium.

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. 2011. Searching Translation Memories for Paraphrases. pages 325–331.

Marcos Zampieri and Mihaela Vela. 2014. Quantifying the Influence of MT Output in the Translators' Performance: A Case Study in Technical Translation. In *Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT 2014)*.