COMPUTER-AIDED TRANSLATION

Preeti[1]

[1]Assistant Professor, Department of Computer Science, IP college, Jhajjar-124103, India.

Abstract- Nowdays , The use of online teaching tools compliments the overall learning and understanding of the subject, and is widely used in schools and universities. Using the web technologies, we create a tool, CaL to help students learn the basics of computation and logic , in parallel with the classroom teaching. The interaction of students with CaL should increase their clarity about the building blocks of logic. According to our research using search engines, there is no such existing tool. The existing tools usually concentrate on higher level concepts such as resolution, natural deduction etc. but CaL, would try build a strong foundation of concepts for the students, useful while learning those higher level concepts. In this paper, I would discuss some of the tools proposed to decrease the translation divergences and try to find out which one proved to be most helpful to the human translators and will look into the underlying details of three types of assistance that are provided to the human translators using various aspects of machine translation. The paper will be concluded by describing what makes the particular tool work best for the translators and how it may be improved by future work in this field.

Keywords- CaL,CAT(computer aided translation),BLEU.

I INTRODUCTION

Computer-aided translation (CAT) uses a machine translation system to speed up the process of human translation, by producing a translation that can be corrected in the post-editing phase by a human translator (Jurafsky and Martin, 2009: 879). Many attempts have been made to make a CAT system independent of human involvement and give a complete and unambiguous translation. Such systems will be a threat to the human translators but would efficiently decrease the overall cost of translating texts. The development of these CAT systems is considered hard due to the systematic, id-iosyncratic and lexical differences, which are referred to as the translation divergences (Jurafsky and Martin, 2009: 880). Given these problems, a fully automatic translation system has not been able to satisfy the professional translators for translating long text documents.

Figure 1.1: Google Translate (https://translate.google.co.in/) from Hindi to English is an example of a Computer-aided translation system, but mostly requires post-editing

Hence, many different approaches have been applied to make the task of translating from one language to another, by providing various assistance tools to the human translators, and not relying completely on the machine translation. This is because the tools have been able to decrease the translation divergences, but could not get rid of them completely; for example the 'Google Translate' as shown in figure 1.1 provides a rough idea of the meaning of the original text but is not semantically correct.

II Types of Assistance

The types of assistance provided to the human translators can be broadly divided into three categories as proposed by Koehn and Haddow (2009), i.e. Sentence Completion Suggestions, Phrase-based Machine Translation and Post-editing Machine Translation.

2.1 Sentence Completion Suggestions

In sentence completion assistance, the user translates the given text word-by-word, and the system interactively makes possible suggestions for the next word or phrase to be added in the translation. This works similarly to the auto-completion functions of email addresses, phone numbers etc. The user can accept the suggestion press-ing an acceptance key or type their translation instead. The system keeps updating the database based on the user input and makes further suggestions by computing the optimal translation of the input sentence.The statistical machine translation system is responsible for making the suggestion to the user. When some 'prefix' of the word is typed by the user, the MT system computes the translation of the input sequence based on the user input and show the most optimal options. Only few words are suggested to the user so that no compromise to the speed is caused by reading long suggestions.
Koehn (2009) created a system (called Caitra) that stored the search graph produced by the machine translation decoder in the database which was used against the user input to calculate the string edit distance between the words. The word with minimum string edit distance and highest sentence translation probability was considered to be an optimal completion path and presented to the user. The system transmitted the entire path to the client but displayed only one phrase prediction at a time, so when the user accepted the suggestion more suggestions could be presented instantly without many computations.

2.2 Phrase-based Machine Translation

Simard and Isabelle (2009) described the phrase-based machine translation system as “a system producing MT output that is consistent with that of a TM when high-similarity material exists in the training data; and providing the MT system with a component that is capable of filtering out machine translations that are less likely to be useful”. The phrase-based MT system is hence based on the translation tables learnt from the vast database of translated texts as proposed by Koehn (2009), or the system computes its translation based on the similarity between the user entered prefix and the translation memory (TM) combined with machine translation, according to Simard and Isabelle (2009).In this approach, each word or word sequence is searched in the translation table to generate the most probable translation options. Koehn’s Caitra (2009), used a heuristic beam search algorithm to find out these options and ordered them accordingly; for the user to decide which will be the correct translation. The user might as well type his/her translation without accepting the suggested phrase. The order of the suggestions is based on the minimum edit distance between the user entered letters and the words in the translation table/memory.

2.3 Post-editing Machine Translation

According to Green et al. (2013) post-editing is the most widely used type of assis-tance, where the human translator manually corrects the fully automated machine translation output. The MT output is generated by the system using the underlying translation memory system or translation tables. The MT output presents a roughly translated version of the text which needs further refinement. So when the user starts translating, the MT output is presented to the user that can be edited as required, im-proving the quality and cost of translation. This reduction in cost has proven to be very beneficial for the industry where there is a need for translation. Hence, the main aim of post-editing is to improve the quality of the machine translated output by correcting the errors

III Tools

Various tools have been created and used to perform this task of translation or provide assistance to the human translators while translating.I outline four of these tools here.

3.1 Caitra

To provide assistance to the translators, Koehn (2009) developed a web-based interac-tive Computer-Aided Translation (CAT) tool, named Caitra which provided all three assistance mentioned in section 2. Caitra was implemented in Ruby as a web-based client-server using Ajax-style Web 2.0 technologies, which was connected to a MySQL database-driven back-end that was powered by Moses decoder (Koehn et al., 2007). Once the document was uploaded using the text box, the back-end pre-computed the required data and presented the user with an interface providing all assistances. These assistances could be turned off by the user as per requirement. The user could translate one sentence at a time given the preceding and the following texts for reference.

Caitra’s interactive sentence completion suggestions were based of TransType (de-scribed in section 3.2) that provided the user with translations of few following words based of the previously translated words. The phrase-based translations were statisti-cally learnt in the form of translation tables from the large amount of translated texts. And the post-editing provided the user with full sentence translations, which were to be corrected by the user. Caitra also tracked the key stroke and mouse click of the user to allow for further analysis of the user and tool interaction, and evaluate the system performance.

3.2 TransType

Langlais et al. (2000) and Macklovitch (2006) described a system called TransType (or TransType2 by Macklovitch (2006)) which made suggestions of next few words for the text, while it was being translated by the human translator. It used an Interactive Machine Translation (IMT) tool that helped in producing high-quality translations. The system worked as a text editor having an embedded MT engine, where the user could select one source text to translate, and begin typing his or her translation. The system proposed its target units according to the characters entered by the user, and kept updating the suggestions while the user was typing. TransType comprised of two parts, namely an evaluator, which assigned probabilis-tic scores to completion hypotheses, and a generator that used the evaluation function to select the best word(s) for completion from an interpolated trigram language model that was trained on the Hansard corpus. The implementation of TransType was based on the object oriented architecture to predict units, from the units that were entered earlier and the source text.
The evaluator was a function p(t |t’, s), which was the conditional probability of the target unit t, given a source text s and the tokens t’ that occurred before t in the current text. The function was defined as in equation 3.1

$$p(t|t',s) = p(t|t')\lambda(\Theta(t',s))+p(t|s)[1-\lambda(\Theta(t',s))] \qquad (3.1)$$

where λ (Θ (t',s)) €[Θ, 1] were the interpolation coefficients and Θ (t',s) was any function that maps t’ and s into a set of equivalence classes. The generator used this function to identify and pick the best unit that matched the current prefix entered by the user. The user could choose to use this proposed unit text in their translation using an acceptance key, modify it or ignore it.

3.3 Translation Memory(TM) Systems

The use of Translation memory (TM) systems along with machine translation (MT) was explored by Lagoudaki (2008), Simard and Isabelle (2009) and Espla` Gomis et al. (2011). Espla` Gomis et al. (2011) noticed that these kinds of systems proved to be very helpful for translators, mainly in case of repetitive texts or texts that focused on a similar topic. The TM system along with MT suggests a translation to the user by searching in the TM resources. When the system cannot find a similar (‘fuzzy’match) or an identical match in the segments of the TM repository it looks at the parts of the segments and finds two sub-segments that form a new segment similar to the source text. According to Lagoudaki (2008), the chance of this segment being correct grammatically is very minimal. But the correction made by the user to this proposed text is stored in the database for future retrieval in case of a similar text encounter.The MT capability of the system to make the correct insertions, matches and trans-lation is a very important issue for the TM systems. The system needs to deal with the overlapping translation examples in the TM database due to the inconsistency present in the translation between two languages. The need to deal with words having different meanings, i.e. word sense disambiguation is also important. The system hence, relies on the statistical scoring to decide which fragments to use. The system also contains a large ‘lexicon’ database; external knowledge of linguistic rules to segment, a corpus with existing translations having source-target pairs. It aligns and does automatic ex-traction from TM content; and annotates the data with grammatical information. The word similarity can then be calculated by measuring the distance between the words, using distance measure metrics such as the Levenshtein distances.Three TM systems have been defined by Lagoudaki (2008), which presented three different aspects of implementation, of TM with MT. An example-based machine translation (EBMT) system, Dej´a` Vu X used statistical scoring to decide which frag-ments to use and, constructed a match if one was not available in the database. It did not make use of the linguistic information, but had another database containing the lexicon of words to complete the sentences adding words like 'for', 'and', etc. EBMT and TM models were compatible due to various similarities between them, which made the im-plementation of this system feasible. Another approach was SIMILIS TM system that applied linguistic rules to processes such as segmentation, alignment and extraction of words from the TM repository. It then segmented the text into sentences and further into smaller 'chunks'; and grammatically annotated the data to attach more informa-tion to each word and indexed them. Hence, the search now looked into each sentence as well as the chunks from the same grammatical category as the source text unit. A slight variation in this system was the Masterin system, which segmented the text ac-cording to the examples present in the TM database. The system annotated the chunks with grammatical information and also attached a grammatical translation pattern to it. The search was done by a pattern recognition method, which was refined by using se-mantics or examining the frequency and domain information. If no match was found, a fuzzy match was constructed using heuristics. To judge the matches, a comparison between the sequences of character strings were used in each system.

Simard and Isabelle (2009) also used the translation memory combined with ma-chine translation as a tool to provide phrase-based assistance. The database of the system was a corpus of existing translation having source-target pairs of sentences and the selected source sentence was called a query. Each query and source sentence was compared using Levenshtein distance, and the pair having least distance was retained. The user could set a threshold for filtering irrelevant translations, which meant in-creasing the recall for precision but at a cost of decreasing the proportion of proposed translations. The Levenshtein distance metric is calculated as in equation 3.2:

$$\mathrm{Sim}(q,s) = \max\left(0, 1 - \frac{\mathrm{levenshtein}(q,s)}{\mathrm{Length}(q)}\right) \qquad (3.2)$$

$$\mathrm{levenshtein}(q,s) = \sum_{i=1}^{N} |q_i - s_i| \qquad (3.3)$$

where q is the query and s is the best-matching source segment in the corpus.

Espla` Gomis et al. (2011) proposed to calculate the fuzzy-match scores (FMS) between the source and target segments as:

$$\mathrm{score}(s'; s_i) = 1 - \frac{D(s', s_i)}{\max(|s'|, |s_i|)} \qquad (3.4)$$

where | z | equals the length in words of the string z and D(z, y) is the Levenshtein distance mentioned above between strings z and y. For the source segment s', the system finds all the translation units (s_i,t_i) having an FMS score above a given thresh-old. This system differed from the others, as it also provided the user with "word-keeping recommendation", where the system automatically recommended which word should be changed and which should be left unedited by the user, hence focusing on the target-side words to change. So those words which are in the translation memory as source-target (s', s_i) pairs can be kept unedited. A set of features can be extracted from a training translation memory and used to translate texts and calculate the scores in advance.

3.4 Deepfix

Rosa et al. (2013) defined a statistical post-editing system named Deepfix, which im-proved the quality of the translated text using syntactic analysis. This system corrected the machine translation output in a second-stage post-processing, hence automatically performing the task of post-editing without human involvement. The translation from one language to another involves a lot of alignment and linguistic knowledge. For ex-ample in Czech, some missing prepositions can change the meaning of the sentences completely, so post-editing is required to correct these errors. Rosa et al. (2013) dealt with the correction of English-to-Czech errors using Deepfix, where the "verb-noun valency", i.e. "the way in which verb and their arguments are used together, and with prepositions and morphological cases".

The use of linguistic knowledge was explored by Rosa et al. (2013), where Tec-togrammatical trees, i.e. "deep syntactic dependency trees based on the Functional Generative Description" showed the relationship between the content words. Every node of the tree corresponded to a lemma of content word such as nouns, verbs or ad-jective along with its attributes; the functional words such as prepositions and auxiliary verbs were added as the attributes of the content words. An example of such tree is shown in figure 3.1:

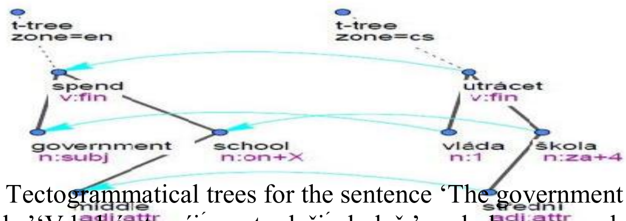

Figure 3.1: Tectogrammatical trees for the sentence 'The government spends on the middle schools.' 'V láda´ utrac´i´ za stredn˘i´ skoly˘:'; only lemmas and formemes of the nodes are shown. (Rosa et al., 2013)

The 'formeme' is the string representation of the selected feature of the content word and its auxiliary words used for the representation of the node. The valency of each verb can be divided into valency frames that describe the meaning of verb along with the arguments that can be attached to the verb. For example, "a valency frame can be expressed as n:subj go n:to+X, which means that the subject goes to some place" (Rosa et al., 2013).

This approach to syntactic analysis followed by post editing of the machine translated text is a very new way of dealing with the task of post-editing and proved to be very helpful tool for the human translators, especially when the translation of similar text is to be done.

IV EVALUATION

The task of evaluation is an essential part of proving the quality of the system created. This can be done either by using human raters or by automatic evaluation such as BLEU (Jurafsky and Martin, 2009) or both. The human raters are considered to be more reliable than automatic evaluation methods, but the later is more cost and time efficient.

4.1 Human Raters

Though most accurate, Human raters tend to be an expensive method of evaluation. Two dimensions, namely fluency and fidelity, are used to evaluate the quality of the translated text. For judging its fluency, human rater is given the translated text and told to rate it on a scale or measure the time taken by the rater to read the given text (clear sentences can be read faster). While the aspects of fidelity that are measured are adequacy (finding if the text contains the information as in the original text) and informativeness (whether the information contained in the translated text is useful for performing some task) (Jurafsky and Martin, 2009). Furthermore, the edit cost of post-editing the machine translated text can be used as an evaluation metric.

4.2 Automatic Evaluation: BLEU

BLEU (Bilingual Evaluation Understudy) was proposed by Papineni et al. (2002) to automatically evaluate the machine translation output. There are many similar heuristic methods, such as NIST, TER, Precision and Recall, and METEOR. These metrics measure the "translation closeness" between the MT output and the human translations(Jurafsky and Martin, 2009). Thus according to Papineni et al. (2002), "the closer a machine translation is to a professional human translation, the better it is". So multiple human translations of each sentence are required to be in the test set for comparison with the MT output.

In BLEU, each MT output is ranked by a weighted average of the number of N-gram overlaps with the human translations. BLEU uses a modified N-gram precision metric, which is computed as in the equations below.

$$p_n = \frac{\sum C \in [Candidates] \sum n{-}gram \in [C] Countclip(n{-}gram)}{\sum C' \in [Candidates] \sum n{-}gram \in [C'] Count(n{-}gram')} \qquad (4.1)$$

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

$$BLEU = BP \times \exp\left(\frac{1}{N} \sum_{n=1}^{N} \log p_n\right) \qquad (4.2)$$

Here, p_n is the precision score which is the N-gram matches for each sentence and the sum of the clipped counts over all the candidate sentences divided by the total number of candidate N-grams in the test set. BP is the brevity penalty for c (the total length of the candidate translation corpus) and r (the effective reference length for that corpus). The equation 4.2 gives the final BLEU value. The higher the value, the better the system performance.

V CONCLUSION

"It is clear that there exists a need for more effective tools that leverage the power of MT, with the flexibility and efficiency of social translation and crowd-sourcing meth-ods" (Eisele et al., 2009). Statistical machine translation has advanced a lot; and more tools are being developed to satisfy the human translators with the perfect translations. The tools we discussed in the report used various machine translation techniques to fulfil the quality criterion, but were not able to provide fully automated MT output that does not require further refinement.

The use of Translation memory (TM) along with MT systems have shown a better performance over other methods, especially when the translated texts are similar in context to the previously translated units. Similarly, the Deepfix tool, which expands the TM systems adding to it the linguistic knowledge and syntactic trees, also proved to be good for the English-to-Czech translation, improving the MT output by more than 56%. This tool can be further expanded to other language pairs using the available parsers. Hence, Deepfix tool can be used as the basis of the future works in this field combining it with other systems for the MT output. But it can be only used as an assistance tool rather than providing the fully-automated translation, which seems to be a futuristic goal of machine translation.

REFERENCES

[1] Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomas,´ J., Vidal, E., et al. (2009). Statistical approaches to computer-assisted translation. Computational Linguistics, 35(1):3–28.

[2] Craciunescu, O., Gerding-Salas, C., and Stringer-O'Keeffe, S. (2004). Machine trans-lation and computer-assisted translation. Translation Journal, 8:3.

[3] Eisele, A., Federmann, C., and Hodson, J. (2009). Towards an effective toolkit for translators. In Proceedings of the ASLIB International Conference Translating and the Computer, volume 31.

[4] Espla` Gomis, M., Sanchez´ Mart´ınez, F., Forcada Zubizarreta, M. L., et al. (2011). Using machine translation in computer-aided translation to suggest the target-side words to change. Machine Translation Summit.

[5] Green, S., Heer, J., and Manning, C. D. (2013). The efficacy of human post-editing for language translation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 439–448. ACM.

[6] Jurafsky, D. and Martin, J. H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, Englewood Cliffs, NJ, 2. ed., [pearson international edition] edition.

[7] Koehn, P. (2009). A web-based interactive computer aided translation tool. In Proceed-ings of the ACL-IJCNLP 2009 Software Demonstrations, pages 17–20. Association for Computational Linguistics.

[8] Koehn, P. and Germann, U. (2014). The impact of machine translation quality on human post-editing. In Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT). Gothenburg, Sweden, pages 38–46.

[9] Koehn, P. and Haddow, B. (2009). Interactive assistance to human translators using statistical machine translation methods. MT Summit XII.

[10] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meet-ing of the ACL on Interactive Poster and Demonstration Sessions, pages 177–180. Association for Computational Linguistics.

[11] Lagoudaki, E. (2008). The value of machine translation for the professional translator. In AMTA-2008. MT at work: Proceedings of the Eighth Conference of the Associ-ation for Machine Translation in the Americas, Waikiki, Hawaii, pages 262–269.

[12] Citeseer.Langlais, P., Foster, G., and Lapalme, G. (2000). Transtype: a computer-aided trans-lation typing system. In Proceedings of the 2000 NAACL-ANLP Workshop on Em-bedded machine translation systems-Volume 5, pages 46–51. Association for Com-putational Linguistics.

[13]Macklovitch, E. (2006). Transtype2: The last word. In Proc. of the conference on International Language Resources and Evaluation, pages 167–17.

[14] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for auto-matic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Com-putational Linguistics.

[15] Rosa, R., Marecek, D., and Tamchyna, A. (2013). Deepfix: Statistical post-editing of statistical machine translation using deep syntactic analysis. In ACL (Student Research Workshop), pages 172–179.

[16]Simard, M. and Isabelle, P. (2009). Phrase-based machine translation in a computer-assisted translation environment. Proceedings of the Twelfth Machine Translation Summit (MT Summit XII), pages 120–127.

JETIR