A Post-editing Interface for Immediate Adaptation in Statistical Machine Translation

Patrick Simianer[†], **Sariya Karimova**[†*], **Stefan Riezler**[†‡]
Department of Computational Linguistics[†] & IWR[‡], Heidelberg University, Germany
[*] Kazan Federal University, Russia
{simianer,karimova,riezler}@cl.uni-heidelberg.de

Abstract

Adaptive machine translation (MT) systems are a promising approach for improving the effectiveness of computer-aided translation (CAT) environments. There is, however, virtually only theoretical work that examines how such a system could be implemented. We present an open source post-editing interface for adaptive statistical MT, which has in-depth monitoring capabilities and excellent expandability, and can facilitate practical studies. To this end, we designed text-based and graphical post-editing interfaces. The graphical interface offers means for displaying and editing a rich view of the MT output. Our translation systems may learn from post-edits using several weight, language model and novel translation model adaptation techniques, in part by exploiting the output of the graphical interface. In a user study we show that using the proposed interface and adaptation methods, reductions in technical effort and time can be achieved.

1 Introduction

Since the earliest beginnings of MT research, it has been obvious to many researchers and practitioners that automatic translation is an outstandingly hard problem and may need human participation for sufficient quality. Accordingly, about 70 years later, systems are not (yet) able to produce perfect, or, depending on the domain, comprehensible translations without human intervention. But, as for example shown by Guerberof (2009), the current quality is sufficient to be used in CAT scenarios, i.e. interactive MT or post-editing. CAT has gained more and more interest from the research community in recent years (Tatsumi, 2010; Koponen, 2016), and now (2016), commercial translation system providers implement and successfully use adaptive MT systems in production[1][2].

Most previous studies in CAT were either evaluated by simulating user behavior or did not consider adaptive translation systems. We seek to conduct studies that examine real user behavior in an adaptive environment. Adapting MT systems to specific users can be advantageous in numerous ways: In simulated experiments of adaptive systems large improvements were shown by taking reference translations as a stand-in for post-edits – significantly reducing the cost of high quality translation; Frustrations, rooted in repeated errors of MT systems, are mitigated and acceptance of MT can be improved; Domain adaptation in MT is capable of greatly improving translation quality; And lastly, translators expect and demand adaptiveness of their tools, as translation memories implement it naturally. To enable studies of user behavior in adaptive environments, we present a post-editing toolkit which can support different types of (adaptive) MT engines, and provide a graphical user interface which includes alignments between sources and their translations. The alignments are extracted from the output of the MT engine and permit novel and precise adaptation methods. Additionally, we provide tools to examine the translation system and the adaptation process, as well as detailed statistics of users' performance in terms of various measures relevant to post-editing.

[1] https://e2f.com/case-study-lilt-travel-portal/
[2] http://blog.translationzone.com/sdl-trados-studio-2017-transformation-translation/

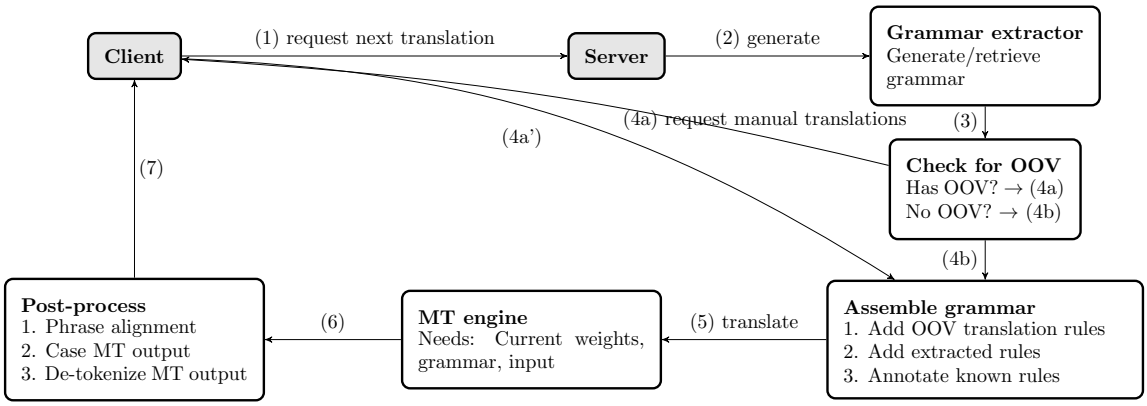

Figure 1: Overview of the steps required to produce a translation that is acceptable for post-editing: After the client (top left) requested a translation of the next segment (1), first the grammar extractor is invoked (2) to produce the grammar (grammars can also be pre-generated), then the input is checked for words that the system is not able to translate (3). If there are untranslatable words, the process returns to the client with a request for translation of the out-of-vocabulary items (OOV) (4a). Then, the grammar can be assembled with all rules that were previously extracted (4a' or 4b). With this grammar, the current weights and the input the MT engine can produce a translation (5), which needs to be processed before it is shown to the user (6, 7).

2 Related work and motivation

Post-editing of MT output is an old idea, going back until the first steps in MT, see e.g. the overview in Koponen (2016). But actual user studies were relatively seldom, as they are expensive to conduct, even more so in the 1960s: The earliest user study to our knowledge describes an offline experiment, which compared comprehensibility of machine translations, post-edits and human made translations (Orr, 1967). With the rise of statistical MT user studies of CAT have gained considerable traction (Casacuberta et al., 2009; Alabau et al., 2013; Federico et al., 2014). Many different toolkits and user interfaces have been used in these studies, for example graphical interfaces for interactive MT specialized for patent translation (Pouliquen et al., 2011), interfaces for predictive translation memories (Green et al., 2014; Koehn, 2009), tools for monitoring post-editing efforts (Aziz et al., 2012), full workbenches supporting post-editing or interactive MT for translators (Alabau et al., 2013; Federico et al., 2014; Casacuberta et al., 2009), or also test-beds for post-editing (Denkowski, 2015). The latter being most similar to our work, even providing a small user study on potential effects of adaptive MT in post-editing.

In most aforementioned interfaces users operate on the string-level and the MT engine is treated as a static black box. Denkowski (2015) is a notable exception, incorporating effective adaptation methods. These, however, also operate only on surface strings, and use a static word alignment model. In contrast, we propose a novel graphical interface that enables efficient and precise resolution of errors in the adaptive MT engine by leveraging user corrected alignments of translation units (e.g. phrases), in conjunction with standard adaptation methods.

3 System overview

Our system can be disassembled into two distinct steps: generation of the output of the MT engine[3], and secondly the adaptation step. The first step is described in Figure 1. The second step, which is comprised of updating the models, is described in Section 5.

[3]Throughout this paper the engine is assumed to be a hierarchical phrase-based SMT engine following Chiang (2007), with a SCFG as the core of its translation model.

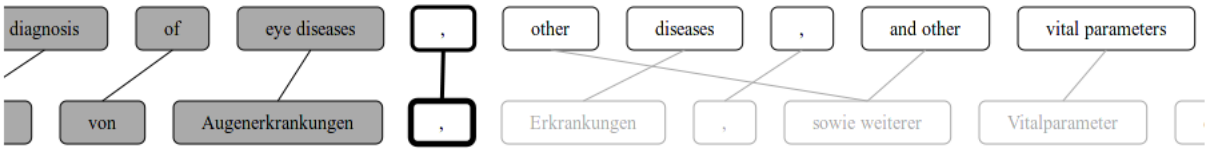

Figure 2: Detail of the graphical user interface, translating from English (top row) to German. Phrases are shown as boxes, alignments are displayed with connecting lines. All target phrases may be interactively moved, deleted, or edited, as well as the alignment links. New phrases can also be added to the target side, only the source side and its segmentation are fixed. The active phrase has a bold border, finished phrases have a dark background.

4 Interfaces

There are two user interfaces implemented: a standard text interface and a graphical interface for enriched presentation of the translation. The text interface consists of two simple input boxes: one contains the source sentence to be translated, the other one is for producing the translation. The target input box can either be pre-filled with a machine translation or left empty for translation from scratch.

In the graphical interface (an example is shown in Figure 2) we show the user not only the target string, but also the latent segmentation and alignment which leads to the translation. The user is invited to not only produce a correct translation, but also to make a sensible alignment of source and target. What is shown can differ between MT approaches: In word-based or current neural MT systems we could simply show the (soft) alignments of words. In phrase-based systems, the source and target are segmented into phrases, and for the hierarchical phrase-based paradigm phrases may be discontinuous, resulting in many-to-many alignments. Our approach not only enables usage of richer structures for adaptation, it has already been shown that visualized word-alignment alone can positively affect users (Schwartz et al., 2015).

From the user-corrected alignment, the MT engine can explicitly learn native corrections to its translation units, which are very valuable compared to updates that only use the surface data (strings). To evaluate and compare different adaptation approaches, the system collects timing information, as well as the number of clicks and keystrokes, all stages of input and output, and weight and model differentials.

5 Adaptation

Our proposed MT engine adaptation partially follows (Denkowski, 2015), using the same adaptive language model and a similar weight adaptation technique, but differs in the adaptation of the translation model. With the graphical interface one is not limited to a simple string pair of source and target translation to adapt the system, but one can also exploit the alignments between the translation units. The chronological sequence of adaptation steps in our implementation is as follows: (1) As shown in Figure 1, OOV is avoided by asking the user for translations of unknown words prior to decoding. (2) After post-editing a sentence, the phrase-segmented and aligned post-edit is compared to the initial machine translation, and lexical corrections and new rules are immediately added to the current grammar. (3) The source is then re-decoded with the augmented grammar, generating a k-best list which is re-ranked by BLEU+1 using the post-edit as reference translation. Weights are updated based on this list with pairwise ranking as described in Simianer et al. (2012). (4) Additionally, we implemented a rule extraction that closely follows the grammar extraction described in Chiang (2007), but instead of words it is using full phrases from the phrase alignment. This results in a large number of additional rules, as all rules (also with gaps) that are compatible with the phrase alignment are extracted for each post-edit provided by the user. To prevent overfitting[4], extracted rules are added to the system in a leave-one-out fashion (i.e. only to subsequent grammars). (5) Before the translation of the next sentence the adaptive language model described by Denkowski (2015) is updated with the string of the current post-edit.

[4]Each translation rule has an associated weight, the same feature set as in Simianer et al. (2012) is used.

response var.	est. Δ	
HBLEU+1	$+6.8 \pm 2.0$ [%]	$p < 0.001$, $\chi^2(1) = 11.748$
HTER	-5.3 ± 1.9 [%]	$p < 0.01$, $\chi^2(1) = 7.8741$
norm. time	-118 ms	—

Table 1: Results of the LMEM analysis. Estimated differences in the response variables contrasting non-adaptive to adaptive systems are given in the Δ column along with their associated p-values, if $p \leq 0.05$. Significance is tested with likelihood ratio tests of the full model against the model without independent variable of interest.

6 User Study

We conducted a user study to test whether our proposed adaptation methods could lead to reduced technical effort or translation speed. For the study we recruited 19 students to use our system in five 90 minute sessions. The group of students was diverse: six study computer science, 13 are prospective translators; the mother tongue of nine students is German, the others were native Italian (7), Spanish, Arabic and Russian (each 1) speakers. The study took place in a controlled environment, all subjects used the same hardware in a computer pool. As translation material we selected patents (Wäschle and Riezler, 2012), where baseline translation performance is good even using smaller, faster models. Since patent claims and descriptions tend to be complex and long, they are not suitable for translation by non-experts. We therefore used titles and abstracts for both training and test. Development and test data are limited to documents with an overall maximum length of 45 tokens per sentence. The data split was done by year and by family id to avoid possible overlaps. Translation direction was English-to-German. The test data were automatically grouped into clusters by cosine similarity of their bag-of-words tf-idf source representations and length, to obtain clusters of related documents with an approximate source token count of 500, which is appropriate in a post-editing setup given the available time limit of 90 minutes. This way, each cluster contained the titles and complete abstracts of 3-5 documents. Two sessions were used to familiarize the subjects with the interface and the translation material using the same task setup as used in the controlled experiments. Each task consists of a document cluster, as described above, which has to be translated within the given time limit. Per session, each cluster is shared by another subject to account for translator variability. Each user uses a dedicated translation system. A session without the proposed adaptation is contrasted to two sessions in which adaptation was enabled. This way, 978 per-sentence measurements were achieved.

Analysis is carried out with linear mixed effects models[5] (LMEM), which are well suited for experimental setups that involve several non-independent measurements, e.g. from multiple responses by the same subjects. Technical translation effort is approximated by HBLEU+1 and HTER, comparing MT outputs to post-edits, and time is normalized by the number of characters in the final post-edits. Raw time cannot be used as response variable since the translation condition (non-adaptive vs. adaptive), the independent variable of interest, is tested with different sets of sources. Random effects (with random intercepts) are subject and source sentence ids, fixed effects are a binary variable separating mother-tongue speakers of German from non-native speakers, and an indicator for source sentence length, which is binned into three distinct levels. Results contrasting the translation condition are given in Table 1. We see significant improvements in HBLEU+1 and HTER, as well as a non-significant time reduction. Quality of post-edits in terms of average BLEU+1 scores with respect to reference translations is stable at 39.6 ± 0.4 [%] across sessions.

7 Conclusions

We presented a toolkit comprised of a text interface, a novel graphical interface and an adaptive MT engine which opens up a wide range of possibilities to carry out interesting post-editing experiments. Our system implements a feedback mechanism to bypass the OOV problem, and with the graphical

[5] Using the implementation of Bates et al. (2012) for R.

user interface, it supports editing of structured MT output which can be leveraged for novel adaptation methods. It also uses and supports existing adaptation techniques, for updating weights, translation and language models. We additionally provide tools to examine the MT engine, as well as the adaptation process, and enable users to evaluate their output in contrast to existing reference translations, and in terms of various measures relevant for post-editing performance. In a user study we could show that adaptive MT engines can significantly reduce technical translation effort in terms of metrics such as HTER or HBLEU+1. The source code is licensed under the LGPL and freely available[6].

References

V. Alabau, R. Bonk, C. Buck, M. Carl, F. Casacuberta, M. G. Martínez, J. González, P. Koehn, L. Leiva, B. Mesa-Lao, D. Ortiz, H. Saint-Amand, G. Sanchis, and C. Tsoukala. 2013. CASMACAT: An open source workbench for advanced computer-aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100.

W. Aziz, S. Castilho, and L. Specia. 2012. Pet: a tool for post-editing and assessing machine translation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.

D. Bates, M. Maechler, and B. Bolker, 2012. *lme4: Linear mixed-effects models using S4 classes*.

F. Casacuberta, J. Civera, E. Cubel, A. L. Lagarda, G. Lapalme, E. Macklovitch, and E. Vidal. 2009. Human interaction for high-quality machine translation. *Communications of the ACM*, 52.

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

M. Denkowski. 2015. *Machine Translation for Human Translators*. Ph.D. thesis, Carnegie Mellon University.

M. Federico, N. Bertoldi, M. Cettolo, M. Negri, M. Turchi, M. Trombetti, A. Cattelan, A. Farina, D. Lupinetti, A. Martines, A. Massidda, H. Schwenk, L. Barrault, F. Blain, P. Koehn, C. Buck, and U. Germann. 2014. The matecat tool. In *Proceedings of the 25th International Conference on Computational Linguistics*.

S. Green, J. Chuang, J. Heer, and C. D. Manning. 2014. Predictive translation memory: A mixed-initiative system for human language translation. In *ACM User Interface Software & Technology*.

A. Guerberof. 2009. Productivity and quality in mt post-editing. In *Proceedings of the MT Summit XII Workshop: Beyond translation memories: New tools for translators*.

P. Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23.

M. Koponen. 2016. Is machine translation post-editing worth the effort? a survey of research into post-editing and effort. *The Journal of Specialised Translation*.

D. B. Orr. 1967. Comprehensibility of machine-aided translations of russian scientific documents.

B. Pouliquen, C. Mazenc, and A. Iorio. 2011. Tapta: A user-driven translation system for patent documents based on domain-aware statistical machine translation. In *Proceedings of the 15th International Conference of the European Association for Machine Translation*.

L. Schwartz, I. Lacruz, and T. Bystrova. 2015. Effects of word alignment visualization on post-editing quality & speed. In *Proceedings of MT Summit XV*.

P. Simianer, S. Riezler, and C. Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

M. Tatsumi. 2010. *Post-Editing Machine Translated Text in A Commercial Setting: Observation and Statistical Analysis*. Ph.D. thesis, Dublin City University.

K. Wäschle and S. Riezler. 2012. Structural and Topical Dimensions in Multi-Task Patent Translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

[6] `https://github.com/pks/lfpe`