IntelliCAT: Intelligent Machine Translation Post-Editing with Quality Estimation and Translation Suggestion

Dongjun Lee[1], **Junhyeong Ahn**[1], **Heesoo Park**[1], **Jaemin Jo**[2]

[1]Bering Lab, Republic of Korea
[2]Sungkyunkwan University, Republic of Korea
{djlee, rkdrnf, heesoo.park}@beringlab.com, jmjo@skku.edu

Abstract

We present IntelliCAT, an interactive translation interface with neural models that streamline the post-editing process on machine translation output. We leverage two quality estimation (QE) models at different granularities: sentence-level QE, to predict the quality of each machine-translated sentence, and word-level QE, to locate the parts of the machine-translated sentence that need correction. Additionally, we introduce a novel translation suggestion model conditioned on both the left and right contexts, providing alternatives for specific words or phrases for correction. Finally, with word alignments, IntelliCAT automatically preserves the original document's styles in the translated document. The experimental results show that post-editing based on the proposed QE and translation suggestions can significantly improve translation quality. Furthermore, a user study reveals that three features provided in IntelliCAT significantly accelerate the post-editing task, achieving a 52.9% speedup in translation time compared to translating from scratch. The interface is publicly available at https://intellicat.beringlab.com/.

1 Introduction

Existing computer-aided translation (CAT) tools incorporate machine translation (MT) in two ways: post-editing (PE) or interactive translation prediction (ITP). PE tools (Federico et al., 2014; Pal et al., 2016) provide a machine-translated document and ask the translator to edit incorrect parts. By contrast, ITP tools (Alabau et al., 2014; Green et al., 2014a; Santy et al., 2019) aim to provide translation suggestions for the next word or phrase given a partial input from the translator. A recent study with human translators revealed that PE was 18.7% faster than ITP in terms of translation time (Green et al., 2014b) and required fewer edits (Do Carmo, 2020). However, many translators still prefer ITP over PE because of (1) high cognitive loads (Koehn, 2009) and (2) the lack of subsegment MT suggestions (Moorkens and O'Brien, 2017) in PE.

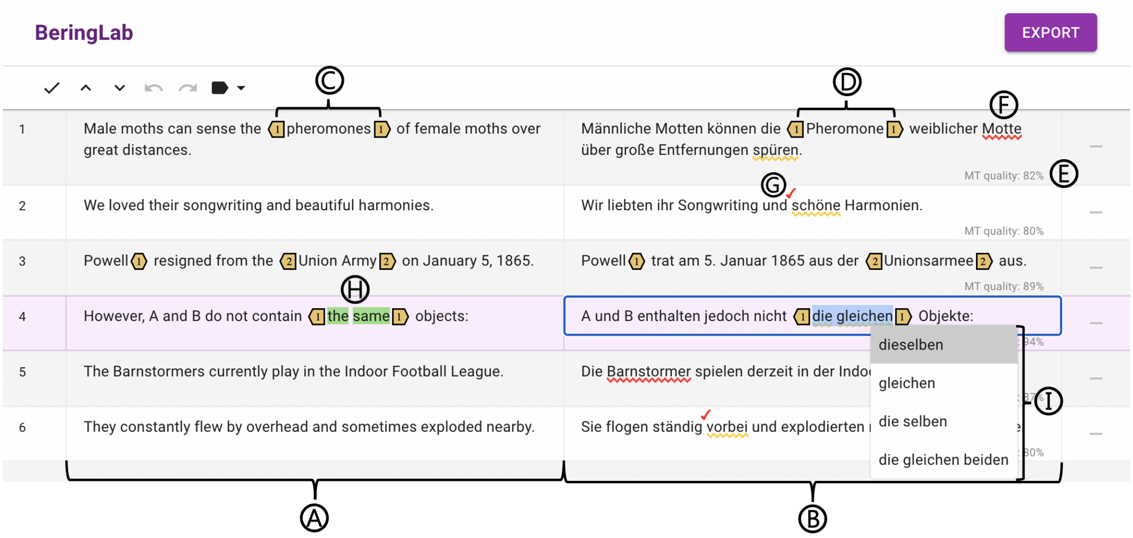
In this paper, we introduce IntelliCAT[1], a hybrid CAT interface designed to provide PE-level efficiency while retaining the advantages of ITP, such as subsegment translation suggestions. To mitigate the cognitive loads of human translators, IntelliCAT aims to automate common post-editing tasks by introducing three intelligent features: (1) quality estimation, (2) translation suggestion, and (3) word alignment.

Quality estimation (QE) is the task of estimating the quality of MT output without reference translations (Specia et al., 2020). We integrate QE into the CAT interface so that the human translator can easily identify which machine-translated sentences and which parts of the sentences require corrections. Furthermore, for words that require post-editing, our interface suggests possible translations to reduce the translators' cognitive load. Finally, based on word alignments, the interface aligns the source and translated documents in terms of formatting by transferring the styles applied in the source document (e.g., bold, hyperlink, footnote, equation) to the translated document to minimize the post-editing time. Our contributions are:

- We integrate state-of-the-art sentence-level and word-level QE (Lee, 2020) techniques into an interactive CAT tool, IntelliCAT.
- We introduce a novel words and phrases suggestion model, which is conditioned on both the left and right contexts, based on XLM-RoBERTa (Conneau et al., 2020). The model is fine-tuned with a modified translation language modeling (TLM) objective (Lample and Conneau, 2019).

[1] A demonstration video is available at https://youtu.be/mDmbdrQE9tc

Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 11–19, August 1st - August 6th, 2021.



1	Male moths can sense the ⟨1 pheromones 1⟩ of female moths over great distances.	Männliche Motten können die ⟨1 Pheromone 1⟩ weiblicher Motte über große Entfernungen spüren.	MT quality: 82%
2	We loved their songwriting and beautiful harmonies.	Wir liebten ihr Songwriting und schöne Harmonien.	MT quality: 80%
3	Powell ⟨1⟩ resigned from the ⟨2 Union Army 2⟩ on January 5, 1865.	Powell ⟨1⟩ trat am 5. Januar 1865 aus der ⟨2 Unionsarmee 2⟩ aus.	MT quality: 89%
4	However, A and B do not contain ⟨1 the same 1⟩ objects:	A und B enthalten jedoch nicht ⟨1 die gleichen 1⟩ Objekte:	94%
5	The Barnstormers currently play in the Indoor Football League.	Die Barnstormer spielen derzeit in der Indo	87%
6	They constantly flew by overhead and sometimes exploded nearby.	Sie flogen ständig vorbei und explodierten	80%

Figure 1: The IntelliCAT Interface. After a document (i.e., an MS Word file) is uploaded, Ⓐ sentences from the original document (source) and Ⓑ the initial MT output for each sentence (target) are shown side-by-side. Ⓒ Formatting tags indicate where a specific style (identified by an integer style id) is applied and Ⓓ are automatically inserted at the proper position of the MT output based on word alignments. Ⓔ The interface shows the quality of each machine-translated sentence based on sentence-level QE. Ⓕ Potentially incorrect words and Ⓖ locations of missing words are highlighted based on word-level QE. When the user selects a sequence of words in the MT output, Ⓗ the corresponding words in the source sentence are highlighted with a heat map, and Ⓘ up to five alternative translations are recommended.

- We conduct quantitative experiments and a user study to evaluate IntelliCAT.

The experimental results on the WMT 2020 English-German QE dataset show that post-editing with the proposed QE and translation suggestion models could significantly improve the translation quality (−6.01 TER and +6.15 BLEU). Moreover, the user study shows that the three features provided by IntelliCAT significantly reduce post-editing time (19.2%), which led to a 52.6% reduction in translation time compared to translating from scratch. Finally, translators evaluate our interface to be highly effective, with a SUS score of 88.61.

2 Related Work

CAT Tool and Post-Editing In the localization industry, the use of CAT tools is a common practice for professional translators (Van den Bergh et al., 2015). As MT has improved substantially in recent years, approaches incorporating MT into CAT tools have been actively researched (Alabau et al., 2014; Federico et al., 2014; Santy et al., 2019; Herbig et al., 2020). One of the approaches is post-editing in which the translator is provided with a machine-translated draft and asked to improve the draft. Recent studies demonstrate that post-editing MT output not only improves translation productivity but also reduces translation errors (Green et al., 2013; Aranberri et al., 2014; Toral et al., 2018).

Translation Suggestion Translation suggestions from interactive translation prediction (ITP) (Alabau et al., 2014; Santy et al., 2019; Coppers et al., 2018) are conditioned only on the left context of the word to be inserted. Therefore, ITP has intrinsic limitations in post-editing tasks where the complete sentence is presented, and the right context of the words that need correction should also be considered. We propose a novel translation suggestion model in which suggestions are conditioned on both the left and right contexts of the words or phrases to be modified or inserted to provide more accurate suggestions when post-editing the complete sentence.

Cross-Lingual Language Model Cross-lingual language models (XLMs), which are language models pre-trained in multiple languages, have led to advances in MT (Lample and Conneau, 2019) and related tasks such as QE (Lee, 2020), automatic post-editing (Wang et al., 2020; Lee et al., 2020), and parallel corpus filtering (Lo and Joanis, 2020). Accordingly, our QE and translation suggestion models are trained on top of XLM-R (Conneau et al., 2020), an XLM that shows state-of-the-art performance for a wide range of cross-lingual tasks. To the best of our knowledge, IntelliCAT is the first CAT interface that leverages XLM to assist human post-editing for MT outputs.

3 System Description

3.1 Overview

IntelliCAT is a web-based interactive interface for post-editing MT outputs (Figure 1). Once loaded, it shows two documents side-by-side: the uploaded original document (an MS Word file) on the left and the machine-translated document on the right. Each document is displayed as a list of sentences with *formatting tags* inserted, tags that show the style of the original document, including text styles (e.g., bold, italic, or hyperlinked) and inline contents (e.g., a media element or an equation).

The user can post-edit MT outputs on the right using the following three features: (1) sentence-level and word-level QE, (2) word or phrase suggestion, and (3) automatic tagging based on word alignments. The sentence-level QE shows the estimated MT quality for each sentence, and word-level QE highlights the parts of each machine-translated sentence that need correction. When the user selects a specific word or phrase, the top-5 recommended alternatives appear below, allowing the user to replace the selected words or insert a new word. Finally, the system automatically captures the original document style and inserts formatting tags in machine-translated sentences at the appropriate locations. After post-editing, the user can click on the export button to download the translated document with the original style preserved. A sample document and its translated document without human post-editing is presented in Appendix A.

3.2 Machine Translation

Our system provides MT for each sentence in the input document. We build our NMT model based on Transformer (Vaswani et al., 2017) using OpenNMT-py (Klein et al., 2017). As training data, the English-German parallel corpus provided in the 2020 News Translation Task (Barrault et al., 2020) is used. We use unigram-LM-based subword segmentation (Kudo, 2018) with a vocabulary size of 32K for English and German, respectively, and the remaining hyperparameters follow the base model of Vaswani et al. (2017).

3.3 Quality Estimation

Quality estimation (QE) is the task of estimating the quality of the MT output, given only the source text (Fonseca et al., 2019). We estimate the quality at two different granularities: sentence and word levels. Sentence-level QE aims to predict the human translation error rate (HTER) (Snover et al., 2006) of a machine-translated sentence, which measures the required amount of human editing to fix the the machine-translated sentence. By contrast, word-level QE aims to predict whether each word in the MT output is `OK` or `BAD` and whether there are missing words between each word.

Figure 1 demonstrates the use of QE in our interface. Based on the sentence-level QE, we show the MT quality for each machine-translated sentence computed as $1 - (predicted\ HTER)$. In addition, based on word-level QE, we show words that need to be corrected (with red or yellow underlines) or locations for missing words (with red or yellow checkmarks). To display the confidence of word-level QE predictions, we encode the predicted probability of the color of underlines and checkmarks (yellow for $P_{BAD} > 0.5$ and red for $P_{BAD} > 0.8$).

For QE training, we use a two-phase cross-lingual language model fine-tuning approach following Lee (2020), which showed the state-of-the-art performance on the WMT 2020 QE Shared Task (Specia et al., 2020). We fine-tune XLM-RoBERTa (Conneau et al., 2020) with a few additional parameters to jointly train sentence-level and word-level QEs. We train our model in two phases. First, we pre-train the model with a large artificially generated QE dataset based on a parallel corpus. Subsequently, we fine-tune the model with the WMT 2020 English-German QE dataset (Specia et al., 2020), which consists of 7,000 triplets consisting of source, MT, and post-edited sentences.

3.4 Translation Suggestion

As shown in Figure 1, when the user selects a specific word or phrase to modify or presses a hotkey (`ALT+s`) between words to insert a missing word, the system suggests the top-5 alternatives based on fine-tuned XLM-R.

XLM-R Fine-Tuning For translation suggestion, we fine-tune XLM-R with a modified translation language modeling (TLM) objective (Lample and Conneau, 2019), which is designed to better predict the masked spans of text in the translation. Following Lample and Conneau (2019), we tokenize source (English) and target (German) sentences with the shared BPE model (Sennrich et al., 2016), and concatenate the source and target tokens with a separation token (`</s>`). Unlike the TLM objective of Lample and Conneau (2019), which randomly masked tokens in both the source and target sentences, we only mask tokens in target sentences since the complete source sentence is always given in the translation task. We randomly replace $p\%$ ($p \in [15, 20, 25]$) of the BPE tokens in the target sentences by `<mask>` tokens and train the model to predict the actual tokens for the masks. In addition, motivated by SpanBERT (Joshi et al., 2020), we always mask complete words instead of sub-word tokens since translation suggestion requires predictions of complete words. As training data, we use the same parallel corpus that is used for MT training.

Inference To suggest alternative translations for the selected sequence of words, we first replace it with multiple `<mask>` tokens. The alternative translations may consist of sub-word tokens of varying lengths. Hence, we generate m inputs, where m denotes the maximum number of masks, and in the i^{th} input ($i \in [1, ..., m]$), the selected sequence is replaced with i consecutive `<mask>` tokens. In other words, we track all cases in which alternative translations consist of 1 to m sub-word tokens. Then, each input is fed into the fine-tuned XLM-R, and `<mask>` tokens are iteratively replaced by the predicted tokens from left to right. In each iteration, we use a beam search with a beam size k to generate the top-k candidates. Finally, all mask prediction results from m inputs are sorted based on probability, and the top-k results are shown to the user.

3.5 Word Alignment and Automatic Formatting

To obtain word alignments, we jointly train the NMT model (§3.2) to produce both translations and alignments following Garg et al. (2019). One attention head on the Transformer's penultimate layer is supervised with an alignment loss to learn the alignments. We use Giza++ (Och and Ney, 2003) alignments as the guided labels for the training. As sub-word segmentation is used to train the NMT model, we convert the sub-word-level alignments back to the word-level. We consider each target word to be aligned with a source word if any of the target sub-words is aligned with the source sub-words.

We provide two features based on word alignment information. First, when the user selects a specific word or phrase in the machine-translated sentence, the corresponding words or phrases in the source sentence are highlighted using a heatmap. Second, formatting tags are automatically inserted at the appropriate locations in the machine-translated sentences. We use two types of tags to represent the formatting of the document: paired tags and unpaired tags. Paired tags represent styles applied across a section of text (e.g., bold or italic). To retain the style applied in the source sentence to the MT, we identify the source word with the highest alignment score for each target word and apply the the corresponding source word's style to the target word. By contrast, unpaired tags represent inline non-text contents such as media elements and equations. To automatically insert an unpaired tag in the MT, we identify the target word with the highest alignment score with the source word right before the tag and insert the corresponding tag after the target word.

4 Experiments

4.1 Model Evaluation

Experimental Setup To evaluate the performance of translation suggestions, we measure MT quality improvement when a sentence is corrected with the suggested words or phrases. We introduce two selection conditions (**Oracle QE** and **Predicted QE**) and two suggestion methods (**XLM-R** and **Proposed**). The selection conditions locate the words that need to be corrected in a sentence; in **Oracle QE** condition, the ground truth word-level QE label is used as a baseline, and in **Predicted QE** condition, our word-level QE model is used to identify the target words. The suggestion methods determine the words that the selected words should be replaced with. We test two suggestion models, the pre-trained XLM-R[2] and the proposed model, fine-tuned with the modified TLM objective, with three different suggestion sizes: top-1, top-3, and top-5.

Each of the QE and translation suggestion models was trained using two Tesla V100 GPUs. As an

[2] https://pytext.readthedocs.io/en/master/xlm_r.html

Model	(With Predicted QE) TER↓	BLEU↑	(With Oracle QE) TER↓	BLEU↑
Baseline (MT)	31.37	50.37	31.37	50.37
XLM-R (Conneau et al., 2020)				
Top-1	30.28 (-1.09)	50.78 (+0.41)	26.57 (-4.80)	56.02(+5.65)
Top-3	29.47 (-1.90)	50.89 (+0.52)	24.10 (-7.27)	60.28 (+9.91)
Top-5	28.75 (-2.62)	51.85 (+1.48)	22.78 (-8.59)	62.40 (+12.03)
Proposed				
Top-1	**29.04 (-2.33)**	**51.93 (+1.56)**	**24.26 (-7.11)**	**59.38 (+9.01)**
Top-3	26.69 (-4.68)	54.70 (+4.33)	19.08 (-12.29)	67.51 (+17.14)
Top-5	25.36 (-6.01)	56.52 (+6.15)	17.30 (-14.07)	70.50 (+20.13)

Table 1: TER and BLEU for machine-translated sentences (Baseline) and post-edited sentences (XLM-R and Proposed) based on word-level QE and translation suggestion.

evaluation dataset, we use the WMT 2020 English-German QE *dev* dataset (Specia et al., 2020). As evaluation metrics, we use the translation error rate (TER) (Snover et al., 2006) and BLEU (Papineni et al., 2002).

Experimental Result Table 1 shows the translation quality of (1) MT sentences (baseline), (2) post-edited sentences with XLM-R-based translation suggestion, and (3) post-edited sentences with the proposed translation suggestion model. When MT sentences are post-edited based on QE prediction with the top-1 suggestion, TER and BLEU are improved over the baseline by -2.33 and $+1.56$, respectively. This result suggests that our QE and translation suggestion models can be used to improve MT performance without human intervention. When the top-5 suggestions are provided, TER and BLEU are improved by -6.01 and $+6.15$, respectively, for the QE prediction condition and improved by -14.07 and $+20.13$, respectively, for the oracle QE condition. These results imply that post-editing based on translation suggestions can significantly improve the translation quality. Finally, the proposed model significantly outperforms XLM-R in all experimental settings, showing that fine-tuning XLM-R with the modified TLM objective is effective for the suggestion performance.

4.2 User Study

We conducted a user study to evaluate the effectiveness of IntelliCAT.

Tasks and Stimuli We asked participants to translate an English document to German using the given interface. As stimuli, we prepared three English documents, each with 12 sentences and 130, 160, and 164 words. The documents included 22, 18, and 20 styles, respectively (e.g., bold, italic, or a footnote), and participants were also asked to apply these styles in the target document.

Translation Interfaces We compared three translation interfaces: **MSWord**, **MT-Only**, and **Full**. In **MSWord**, the participants were asked to translate documents using a popular word processor, Microsoft Word. In this baseline condition, two Microsoft Word instances were shown side-by-side: one showing an English document (source) and the other showing an empty document where one could type the translated sentences (target). In **MT-Only**, participants started with a machine-translated document on IntelliCAT without QE, translation suggestion, and word alignment; they had to edit incorrect parts and transfer styles by themselves. In **Full**, the participants could use all the features of IntelliCAT.

Participants and Study Design We recruited nine participants (aged 23–31 years). All participants majored in German and were fluent in both English and German. We adopted a within-subject design; each participant tested all three interfaces and three documents. Thus, our study consisted of nine (participants) × 3 (conditions) = 27 trials in total. The order of interfaces and documents was counterbalanced using a 3 × 3 Latin square to alleviate the possible bias of learning effects or fatigue. For each trial, we measured the translation completion time.

Procedure Participants attended a training session for ten minutes, where they tried each interface with a short sample document. Subsequently, they performed three translation tasks with different interfaces. We allowed them to look up words for which they did not know the translation before

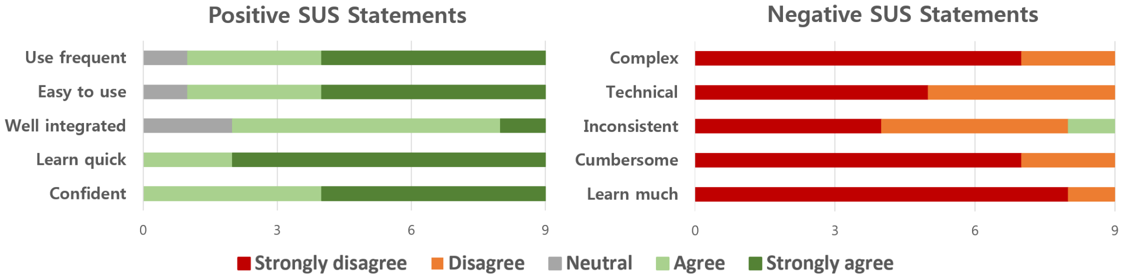


Figure 2: SUS Feedback. The usability of IntelliCAT was evaluated as an excellent level with a score of 88.61±7.82.

starting each translation task. Upon completing the three tasks, participants responded to a system usability scale (SUS) questionnaire (Brooke, 1996), and we gathered subjective feedback. The entire session took approximately 90 min per participant.

Interface	Avg. time (s)
MSWord	1178.78 ± 280.41
MT-Only	688.00 ± 175.02
Full	$\mathbf{555.66 \pm 200.81}$

Table 2: Translation completion time. The differences between the three interface conditions are statistically significant.

Result and Discussion Table 2 summarizes the result of the user study. A repeated measures ANOVA with a Greenhouse-Geisser correction found a significant difference in completion time between the three translation interfaces ($F(1.306, 10.449) = 56.398$, $p < 0.001$). Post hoc tests using the Bonferroni correction revealed that **Full** (555.66 ± 200.81 s) was significantly faster than **MT-Only** (688.00 ± 175.02 s) ($p = 0.013$) and **MT-Only** was significantly faster than **MSWord** (1,178.78 ± 280.41 s) ($p < 0.001$). These results suggest that our QE, translation suggestion, and word alignment features could further accelerate post-editing (a 19.2% speedup) (**Full** vs. **MT-Only**), and our system could reduce the translation time by more than half (52.9%) compared to translating from scratch (**Full** vs. **MSWord**).

We could not find a significant difference between documents ($F(1.964, 15.712) = 0.430$, ns) with the same statistical procedure, which suggests that the translation difficulties of the three English documents were not statistically different.

Our interface received a mean SUS score of 88.61 ($\sigma = 7.82$), which is slightly higher than the score for an “Excellent” adjective ratings (85.58, Bangor et al. (2008)). Eight out of nine participants reported that QE was useful for proofreading purposes; P2 stated, “With QE, I could double-check the words that are possibly wrong.” All participants evaluated the translation suggestions to be useful; P7 mentioned “Translation suggestion was very convenient. It might significantly reduce the dependence on the dictionary.”

Overall, the user study results demonstrated the effectiveness of IntelliCAT both quantitatively and qualitatively, and we found that human translators could streamline their post-editing process with the three features provided in IntelliCAT.

5 Conclusion and Future Work

In this paper, we introduce IntelliCAT, an intelligent MT post-editing interface for document translation. The interface provides three neural network-based features to assist post-editing: (1) sentence-level and word-level QEs, (2) alternative translation suggestions for words or phrases, and (3) automatic formatting of the translated document based on word alignments. The model evaluation shows that post-editing based on the proposed QE and translation suggestion models can significantly improve the quality of translation. Moreover, the user study shows that these features significantly accelerate post-editing, achieving a 52.9% speedup in translation time compared to translating from scratch. Finally, the usability of IntelliCAT was evaluated as an “excellent” level, with a SUS score of 88.61.

In future work, we will build a pipeline that continuously improves the performance of neural models based on automatically collected triplets consisting of source, MT, and post-edited sentences. We will implement an automatic post-editing (Chatterjee et al., 2020) model to continuously improve MT performance and apply online learning to QE models to continually enhance QE performance.

References

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis A Leiva, et al. 2014. Casmacat: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28.

Nora Aranberri, Gorka Labaka, A Diaz de Ilarraza, and Kepa Sarasola. 2014. Comparison of post-editing productivity between professional translators and lay users. In *Proceeding of AMTA Third Workshop on Post-editing Technology and Practice (WPTP-3), Vancouver, Canada*, pages 20–33.

Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, et al. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55.

Jan Van den Bergh, Eva Geurts, Donald Degraen, Mieke Haesen, Iulianna Van der Lek-Ciudin, Karin Coninx, et al. 2015. Recommendations for translation environments to improve translators' workflows. *Translating and the Computer*, 37:106–119.

John Brooke. 1996. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the wmt 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Sven Coppers, Jan Van den Bergh, Kris Luyten, Karin Coninx, Iulianna Van der Lek-Ciudin, Tom Vanallemeersch, and Vincent Vandeghinste. 2018. Intellingo: An intelligible translation environment. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Félix Do Carmo. 2020. Comparing post-editing based on four editing actions against translating with an auto-complete feature. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 421–430.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, et al. 2014. The matecat tool. In *COLING (Demos)*, pages 129–132.

Erick Fonseca, Lisa Yankovskaya, André FT Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4443–4452.

Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D Manning. 2014a. Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 177–187.

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*, pages 439–448.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014b. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Nico Herbig, Tim Düwel, Santanu Pal, Kalliopi Meladaki, Mahsa Monshizadeh, Antonio Krüger, and Josef van Genabith. 2020. Mmpe: A multi-modal interface for post-editing machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1691–1702.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Gil Kim, and Jong-Hyeok Lee. 2020. Postech-etri's submission to the wmt2020 ape shared task: Automatic post-editing with cross-lingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 777–782.

Chi-kiu Lo and Eric Joanis. 2020. Improving parallel data identification using iteratively refined sentence alignments and bilingual mappings of pre-trained language models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 972–978.

Joss Moorkens and Sharon O'Brien. 2017. Assessing user interface needs of post-editors of machine translation. *Human issues in translation technology*, pages 109–130.

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

Santanu Pal, Marcos Zampieri, Sudip Kumar Naskar, Tapas Nayak, Mihaela Vela, and Josef van Genabith. 2016. Catalog online: Porting a post-editing tool to the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 599–604.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmén, and André F. T. Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

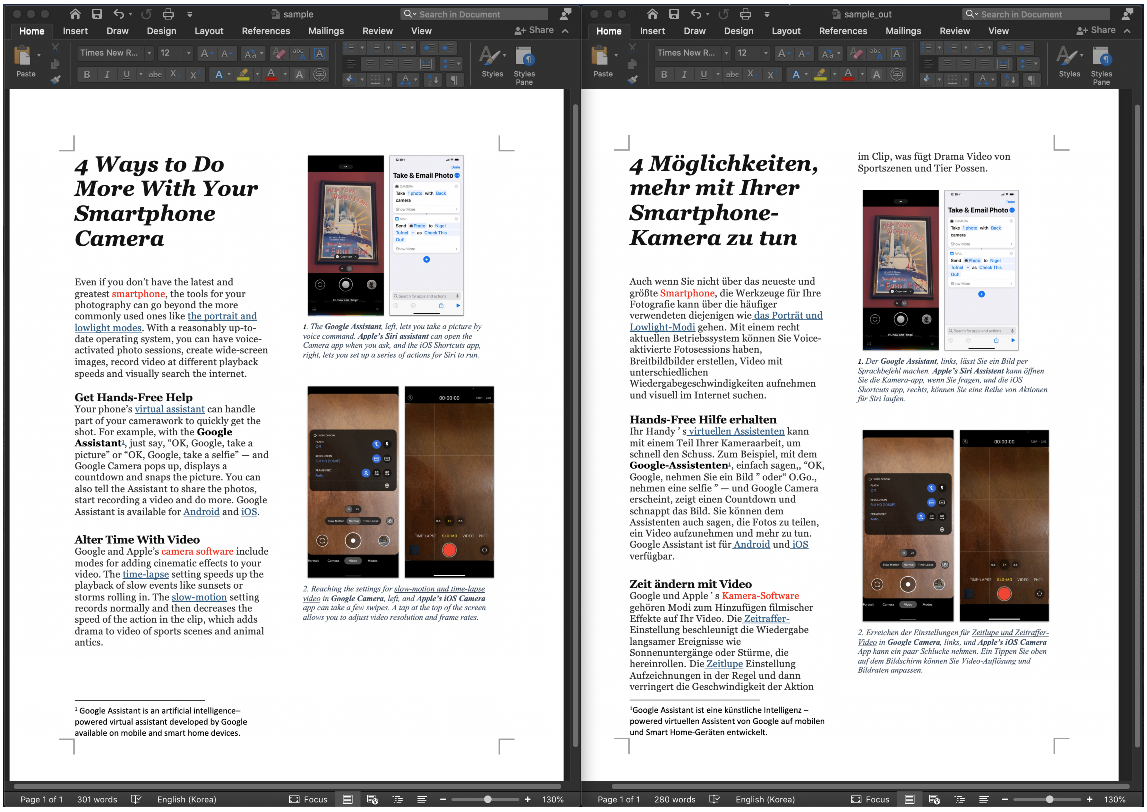
Antonio Toral, Martijn Wieling, and Andy Way. 2018. Post-editing effort of a novel with statistical and neural machine translation. *Frontiers in Digital Humanities*, 5:9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jiayi Wang, Ke Wang, Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi, and Yu Zhao. 2020. Alibaba's submission for the wmt 2020 ape shared task: Improving automatic post-editing with pre-trained conditional cross-lingual bert. In *Proceedings of the Fifth Conference on Machine Translation*, pages 789–796.

A Sample Document Translation

Figure 3 shows a sample document and the translated document using IntelliCAT without human intervention.



4 Ways to Do More With Your Smartphone Camera

Even if you don't have the latest and greatest smartphone, the tools for your photography can go beyond the more commonly used ones like the portrait and lowlight modes. With a reasonably up-to-date operating system, you can have voice-activated photo sessions, create wide-screen images, record video at different playback speeds and visually search the internet.

Get Hands-Free Help

Your phone's virtual assistant can handle part of your camerawork to quickly get the shot. For example, with the **Google Assistant**[1], just say, "OK, Google, take a picture" or "OK, Google, take a selfie" — and Google Camera pops up, displays a countdown and snaps the picture. You can also tell the Assistant to share the photos, start recording a video and do more. Google Assistant is available for Android and iOS.

Alter Time With Video

Google and Apple's camera software include modes for adding cinematic effects to your video. The time-lapse setting speeds up the playback of slow events like sunsets or storms rolling in. The slow-motion setting records normally and then decreases the speed of the action in the clip, which adds drama to video of sports scenes and animal antics.

Take & Email Photo
Done
CAMERA
Take 1 photo with Back camera
Show More
MAIL
Send Photo to Nigel Tufnel + as Check This Out!
Show More
Search for apps and actions
Copy text
Hi, how can I help?

*1. The **Google Assistant**, left, lets you take a picture by voice command. **Apple's Siri assistant** can open the Camera app when you ask, and the iOS Shortcuts app, right, lets you set up a series of actions for Siri to run.*

*2. Reaching the settings for slow-motion and time-lapse video in **Google Camera**, left, and **Apple's iOS Camera** app can take a few swipes. A tap at the top of the screen allows you to adjust video resolution and frame rates.*

[1] Google Assistant is an artificial intelligence–powered virtual assistant developed by Google available on mobile and smart home devices.

4 Möglichkeiten, mehr mit Ihrer Smartphone-Kamera zu tun

Auch wenn Sie nicht über das neueste und größte Smartphone, die Werkzeuge für Ihre Fotografie kann über die häufiger verwendeten diejenigen wie das Porträt und Lowlight-Modi gehen. Mit einem recht aktuellen Betriebssystem können Sie Voice-aktivierte Fotosessions haben, Breitbildbilder erstellen, Video mit unterschiedlichen Wiedergabegeschwindigkeiten aufnehmen und visuell im Internet suchen.

Hands-Free Hilfe erhalten

Ihr Handy ' s virtuellen Assistenten kann mit einem Teil Ihrer Kameraarbeit, um schnell den Schuss. Zum Beispiel, mit dem **Google-Assistenten**[1], einfach sagen,, "OK, Google, nehmen Sie ein Bild " oder" O.Go., nehmen eine selfie " — und Google Camera erscheint, zeigt einen Countdown und schnappt das Bild. Sie können dem Assistenten auch sagen, die Fotos zu teilen, ein Video aufzunehmen und mehr zu tun. Google Assistant ist für Android und iOS verfügbar.

Zeit ändern mit Video

Google und Apple ' s Kamera-Software gehören Modi zum Hinzufügen filmischer Effekte auf Ihr Video. Die Zeitraffer-Einstellung beschleunigt die Wiedergabe langsamer Ereignisse wie Sonnenuntergänge oder Stürme, die hereinrollen. Die Zeitlupe Einstellung Aufzeichnungen in der Regel und dann verringert die Geschwindigkeit der Aktion im Clip, was fügt Drama Video von Sportszenen und Tier Possen.

*1. Der **Google Assistant**, links, lässt Sie ein Bild per Sprachbefehl machen. **Apple's Siri Assistent** kann öffnen Sie die Kamera-app, wenn Sie fragen, und die iOS Shortcuts app, rechts, können Sie eine Reihe von Aktionen für Siri laufen.*

*2. Erreichen der Einstellungen für Zeitlupe und Zeitraffer-Video in **Google Camera**, links, und **Apple's iOS Camera** App kann ein paar Schlucke nehmen. Ein Tippen Sie oben auf dem Bildschirm können Sie Video-Auflösung und Bildraten anpassen.*

[1] Google Assistant ist eine künstliche Intelligenz – powered virtuellen Assistent von Google auf mobilen und Smart Home-Geräten entwickelt.

Figure 3: A sample document (left) and the translated document (right) without human intervention.