Machine Translation (2019) 33:135–154
https://doi.org/10.1007/s10590-019-09235-8

A user study of neural interactive translation prediction

Rebecca Knowles[1] · **Marina Sanchez-Torron**[2] · **Philipp Koehn**[1,3]

Received: 13 July 2018 / Accepted: 22 January 2019 / Published online: 2 May 2019

Abstract
Machine translation (MT) on its own is generally not good enough to produce high-quality translations, so it is common to have humans intervening in the translation process to improve MT output. A typical intervention is post-editing (PE), where a human translator corrects errors in the MT output. Another is interactive translation prediction (ITP), which involves an MT system presenting a translator with translation suggestions they can accept or reject, actions the MT system then uses to present them with new, corrected suggestions. Both Macklovitch (2006) and Koehn (2009) found ITP to be an efficient alternative to unassisted translation in terms of processing time. So far, phrase-based statistical ITP has not yet proven to be faster than PE (Koehn 2009; Sanchis-Trilles et al. 2014; Underwood et al. 2014; Green et al. 2014; Alves et al. 2016; Alabau et al. 2016). In this paper we present the results of an empirical study on translation productivity in ITP with an underlying neural MT system (NITP). Our results show that over half of the professional translators in our study translated faster with NITP compared to PE, and most preferred it over PE. We also examine differences between PE and ITP in other translation productivity indicators and translators' reactions to the technology.

Keywords Computer aided translation · Machine translation · User study

Rebecca Knowles and Marina Sanchez-Torron contributed equally to this work

✉ Philipp Koehn
phi@jhu.edu

Rebecca Knowles
rknowles@jhu.edu

Marina Sanchez-Torron
msnc017@aucklanduni.ac.nz

1 Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

2 School of Cultures, Languages and Linguistics, University of Auckland, Auckland 1010, New Zealand

3 School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, Scotland EH8-9AB, UK

1 Introduction

Interactive translation prediction (ITP) serves to allow translators to work with the output of machine translation (MT) systems by using it like an "auto-complete" feature. Rather than starting with a complete (but likely erroneous) translation which they then must post-edit (PE), a translator using ITP guides the translation process. They can accept a suggestion with a single keystroke, or reject it by typing an alternate translation. When a suggestion is rejected, the MT system recomputes its predictions from the given prefix and presents its new suggestions to the translator.

As described in Wuebker et al. (2016) and Knowles and Koehn (2016), this is done in *neural* interactive translation prediction (NITP) by feeding the translator's token(s) into the neural machine translation (NMT) model as conditioning context (rather than feeding in the rejected system predictions), then producing the rest of the translation token by token. Using reference text to simulate translators, both papers show that NITP outperforms ITP systems that are based on phrase-based statistical MT even when the underlying MT systems are of similar quality.

In this work, we investigate the use of NITP through a user study with professional English-Spanish translators. We integrate an NITP system into a web-based translation workbench (Fig. 1) and conduct a user study with eight professional translators. We find that most translators in our study prefer NITP to PE, and most would be willing to use it in their work. Over half of the translators translated faster with NITP than PE, but we do not find a significant difference between translation speed with NITP and PE overall. We provide some analysis of translator reactions to the tool, including a discussion of the potential relationship between translator experience with PE and their reactions to ITP assistance.

2 Related work

Our work focuses on neural interactive translation prediction. However, the earliest body of work on ITP, including the TransType and TransType2 projects (Langlais et al. 2000; Foster et al. 2002; Bender et al. 2005; Barrachina et al. 2009), predates the current wave of neural approaches to MT. Following those projects, approaches using static search graphs were proposed to allow for ITP from phrase-based statistical MT. In the search graph approach, the system seeks to find a match for the prefix (the partial translator input) in the search graph, backing off to edit-distance techniques when exact matches are not found (Koehn 2009; Koehn et al. 2014). An alternative approach for statistical phrase-based MT is to use constrained decoding (Wuebker et al. 2016). The proposed approach for NITP (described in more detail in the following sections) was introduced in Wuebker et al. (2016) and Knowles and Koehn (2016), who found in simulations that NITP outperformed phrase-based statistical ITP approaches, in terms of their accuracy in predicting the next word after a translator-generated prefix. Some computer-aided translation workbenches, including the research environment

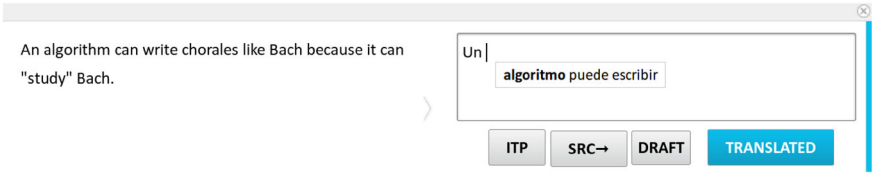

Fig. 1 Interactive translation prediction in CASMACAT: The system suggests continuing the translation with *algoritmo puede escribir*, which the user can accept by pressing the TAB key

CASMACAT[1] (see Fig. 1) and the commercial tool Lilt[2] contain implementations of ITP.

The effect of ITP on translation productivity has been assessed through simulations and empirical studies, with much of the focus placed on processing time and technical effort (relative to unassisted translation or PE). Macklovitch (2006) found that the TransType2 ITP research system increased translators' productivity in terms of processing time (relative to unassisted translation) by about 15–20%, and produced texts of comparable quality, while Barrachina et al. (2009), in a simulated setting, showed that ITP had the potential to reduce typing effort between about 55% and 80%. Koehn (2009) found that, relative to unassisted translation, both ITP and PE produced better quality, faster translations, but that ITP did not yet yield time gains to the level of PE. Similar findings of translators being slower overall in ITP than in PE are reported in Underwood et al. (2014), Green et al. (2014), Sanchis-Trilles et al. (2014), Alabau et al. (2016) and Alves et al. (2016). The number of participants in these research studies ranged from 5 to 32; language pairs investigated were English to Spanish, Portuguese, and German, and French to English. Except for Green et al. (2014), all studies were conducted on CASMACAT. Findings on the keystroke activity involved in ITP, however, are somewhat contradictory, with Sanchis-Trilles et al. (2014) finding it to be lower in ITP, and Alves et al. (2016) obtaining the opposite result. This is likely due to differences in how the interactive functionality was implemented, with the former producing a translation of the entire sentence—instead of word-by-word suggestions that need to be confirmed—every time a keystroke was made. Findings on cognitive effort are also mixed, with Alves et al. (2016) reporting that ITP involved more gaze fixation counts than PE, but that their total duration was lower, and Underwood et al. (2014) reporting that gaze duration across conditions was similar, though more gaze attention was placed on the target text than on the source text in the ITP condition. As for final translation quality, Alves et al. (2016) and Underwood et al. (2014) found ITP and PE to result in comparable quality measured in terms of edit distance, while Green et al. (2014) found that translations done using ITP yielded slightly higher BLEU scores than those done in PE. Finally, in terms of translator satisfaction, and unlike Macklovitch (2006) or Underwood et al. (2014), Koehn (2009) found that, overall, translators preferred ITP over PE.

[1] http://www.casmacat.eu.

[2] https://lilt.com/.

2.1 Neural machine translation

Here we focus on an encoder-decoder with attention (one commonly used neural architecture), as described in Bahdanau et al. (2015) and implemented in the Nematus NMT tool (Sennrich et al. 2017). We use byte-pair encoding (BPE; Sennrich et al. 2016) to perform translation at the subword level. In the encoder, the preprocessed input sentence is passed through two recurrent neural networks (one left-to-right, one right-to-left), which are then concatenated together such that the hidden state (h_t) associated with each input token (x_t) contains information about that token and its full input sentence context. The decoder produces the output sentence one token at a time (left to right), conditioned on the previously produced tokens and an attention mechanism, which serves as a soft alignment to the representations of the input.

At step t, the conditional probability of generating output token y_t given the full input sequence $\mathbf{x}$ and the previously output tokens $\hat{y}_1, \ldots, \hat{y}_{t-1}$ is:

$$p(y_t|\{\hat{y}_1, \ldots, \hat{y}_{t-1}\}, \mathbf{x}) = g(\hat{y}_{t-1}, c_t, s_t) \tag{1}$$

where g is a non-linearity and c_t and s_t are the context vector and hidden state, respectively. The vector c_t is a weighted average of all encoder hidden states h_j, with weights generated by the attention mechanism.

2.2 Neural interactive translation prediction

In NITP, instead of conditioning the prediction of each token on the previous model predictions $\{\hat{y}_1, \ldots, \hat{y}_{t-1}\}$ (as is done in standard NMT decoding), we condition on the true translator-generated prefix $\{y^*_1, \ldots, y^*_{t-1}\}$. This results in a new conditional probability equation:

$$p(y_t|\{y^*_1, \ldots, y^*_{t-1}\}, \mathbf{x}) = g(y^*_{t-1}, c_t, s_t) \tag{2}$$

That is, the conditioning context is now the one produced by the human translator rather than the one produced by the MT system's predictions. In practice, we generate more than just the next predicted token to show to the translator, so it is better described as follows: given a translator prefix of length m, and some number n of next tokens which we wish to show to the translator, we have two equations.

$$p(y_{m+1}|\{y^*_1, \cdots, y^*_m\}, \mathbf{x}) = g(y^*_m, c_t, s_t) \tag{3}$$

$$p(y_{m+n}|\{y^*_1, \cdots, y^*_m, \hat{y}_{m+1}, \cdots, \hat{y}_{m+n-1}\}, \mathbf{x}) = g(\hat{y}_{m+n-1}, c_t, s_t) \forall n > 1 \tag{4}$$

In Eq. 3, we see that the word immediately following the user-generated prefix is conditioned on the user-generated prefix. In Eq. 4, we see that all subsequent words are conditioned on a user-generated prefix followed by predicted words (until such time as the translator accepts or rejects them).

If a translator rejects a suggestion and provides their own, there are two possible cases: either the translator has added a complete word to the translation, or they have

added a partial word. In the case of a complete word, we follow Eqs. 3 and 4. That word becomes part of the prefix, and the generation of the subsequent tokens is conditioned on it.

If, however, the translator has only generated a partial word (which we will call a character prefix), this is slightly more complicated. We provide some additional technical detail here. We must first determine whether this character prefix is the prefix to any item in our (subword) vocabulary. If it is the prefix of at least one vocabulary item, we predict the completion to this word (or subword) by selecting the highest probability item in the vocabulary that starts with our character prefix (this can be described as a modification to the softmax and/or as a mask applied to the distribution prior to performing the softmax). Given the character prefix r^*:

$$p(y_t|\{y_1^*,\ldots,y_{t-1}^*,r^*\},\mathbf{x}) \propto \mathbb{1}(y_t)p(y_t|\{y_1^*,\ldots,y_{t-1}^*\},\mathbf{x}) \tag{5}$$

where

$$\mathbb{1}(y_t) = \begin{cases} 1 & \text{if } y_t \text{ starts with the string } r^* \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We then continue predicting the remaining tokens in the standard fashion.

In the case that the character prefix is *not* the prefix to any item in our vocabulary, we must first apply BPE to it.[3] Once BPE has been applied, we have the model consume (forced decode) all but the last subsegment. This last subsegment could be a complete vocabulary item on its own, or again a prefix to a vocabulary item. Thus we return to our approach of predicting the highest probability vocabulary item which has the last subsegment as a prefix, and then continue prediction.

This approach, as described in the Letter Prediction Accuracy section of Knowles and Koehn (2016), eliminates the need to further modify our decoder, while maintaining the character-level interactions expected in ITP. Knowles and Koehn (2016) also propose speed-related improvements, which we discuss in Appendix B.

3 NITP system and study setup

We integrated an implementation of NITP based on Nematus into the open-source CASMACAT translation workbench (Alabau et al. 2014), which uses a similar layout and keyboard combinations to many commercial CAT tools (albeit without some common features like spell check, integrated dictionary or concordancer functionality) and which was also used in a number of the studies described in Sect. 2. We then conducted a longitudinal empirical study with a threefold purpose: (a) comparing translation productivity in ITP with that of PE; (b) investigating whether translation productivity in ITP improved as translators became familiar with the ITP technology, and (c) collecting translators' impressions of ITP.

[3] This has potentially interesting consequences, as the BPE segments produced here may not be the ones that would have been produced had the translator produced the entire vocabulary item in one go.

We trained an English–Spanish NMT model using the attention-based encoder-decoder toolkit Nematus (Sennrich et al. 2017).[4] We preprocessed the data using the standard preprocessing scripts: tokenization, truecasing, and BPE (Sennrich et al. 2016). The system is trained on Europarl v7 (Koehn 2005) and News Commentary v10 data,[5] which comprised the WMT13 training data for English–Spanish. This training set contains 3.95 million sentence pairs, over 102 million source tokens, and over 106 million target tokens. We used the WMT12 News Test data for validation. The system has a BLEU score of 29.79 (beam 12, less than 1 BLEU below the best score from WMT13) or 28.40 (beam 1) on the WMT13 test set and a reference-simulated word prediction accuracy of 59.1% (beam 1).[6]

Eight Castilian Spanish professional translators (referred to as TrA through TrH) participated in the study. The original sample size was reduced by about 17% due to technical (server down) issues invalidating two translator-session combinations and to TrB producing unusable translation activity data by not adhering to instructions (we nevertheless report data on TrB's background and the feedback provided by him as they may help put said nonadherence in context—see Sect. 5.3 for more details). The study consisted of eight sessions spanning four weeks; in the first session, translators engaged in PE (N=201 sentences); in the next seven sessions, they engaged in ITP (N=1349). From the first session we obtained a PE baseline against which we compared translation productivity in the ITP setting; potential learning effects derived from repeated ITP sessions were assessed by examining the indicators collected in the remaining seven sessions.

Eight news texts, controlled for length and syntactic complexity,[7] were selected for the user study. They dealt with a range of topics like politics, technology, business, and life and style. Texts had on average 29.13 sentences ($SD = 1.24$), 822.75 tokens ($SD = 37.48$), and a dependency length of 103 ($SD = 2.99$) and were assigned randomly to translators, while ensuring that each text was presented only once in each session and only once to each translator throughout the study. Translators were asked to produce publication quality translations with two specific guidelines: (1) use as much of the MT output as possible, as in Massardo et al. (2011), and (2) do not engage in preferential changes that do not improve the quality of the text.

[4] We use these training parameters: vocabulary of size 50,000, word embedding layer size of 500, hidden layer size of 1000, batch size of 80, Adadelta optimizer (Zeiler 2012), maximum sentence length of 50, and default learning rate of 0.0001. All other parameters are set to Nematus defaults.

[5] http://www.casmacat.eu/corpus/news-commentary.html.

[6] Word Prediction Accuracy (WPA) is the percentage of words that the NITP system predicted correctly, given a prefix of all the previous reference/translator-produced words.

[7] As a proxy for translation difficulty, we measure syntactic complexity using the length of dependency links in the dependency structure of the sentence (an approach proposed in Lin (1996), validated in Mishra et al. (2013) and used in Green et al. (2013). Specifically, basic dependencies, with punctuation relationships included, were obtained with the Stanford CoreNLP Natural Language Processing Toolkit (Manning et al. 2014).

3.1 Translator interactions

Translators were provided with detailed instructions about the study, including compensation, interaction modes, and translation quality expectations, via participant information sheets and a help page (see Appendix A). Translators conducted a warm-up task consisting of PE and interactively translating five sentences prior to the main task, to make sure they familiarized themselves with the translation environment and with the interaction modes.

The CASMACAT system logs all keystrokes, mouse clicks, and movements between segments in the interface, along with timestamps. The system also logs requests to the translation server, source data, initial translation data, and final translation output produced by the translators. While the underlying translation system vocabulary consists of subword segments, user interactions are performed at the character level (by typing individual characters) and at the whole-word level (by hitting TAB to accept a suggestion). All byte-pair operations are performed behind the scenes and are not shown to the user.

In the user interface (UI), shown in Fig. 1 in ITP mode, translators see a source sentence on the left and a space to enter their translation on the right. They translate the document sentence-by-sentence. During PE, the right side is initially populated with MT output, which the translator then edits, as in a standard word processor. During ITP, a floating box to the right and below the translator-produced prefix shows the next three suggested words. The translator can accept a word using the TAB key, or type a new word.

4 Operationalization

Translation productivity was measured through eleven variables in three categories: temporal effort (Processing Time), technical effort (Manual Insertions, Manual Deletions, Navigation and Special Key Presses, Mouse Clicks, and Tokens of MT Origin) and final translation quality (MQM Score,[8] Accuracy Issues, Fluency Issues, Minor Issues, and Major Issues). More specifically, Processing Time was measured in seconds; Manual Insertions and Manual Deletions were the count of alphanumeric characters manually inserted or deleted, respectively, by the translator; Navigation and Special Key Presses were the count of navigation (UP, DOWN, LEFT and RIGHT), control (CTRL, ALT, SHIFT) and TAB key presses. Mouse usage was measured via the count of Mouse Clicks. Tokens of MT Origin measured the count of tokens in the final, target text that were accepted by the translators exactly and not changed after being accepted as suggested by the ITP system, or, in the case of PE, that were left unedited (i.e., not altered or moved around). Lastly, the MQM manual annotation framework allowed us to assign to each post-edited/interactively translated sentence a measure of translation quality defined via: (a) an MQM Score (0–100%) according to which a Pass ($\geq$ 95%) or a Fail ($<$ 95%) status was assigned and (b) the frequency of issues, classified according to their type (Accuracy and Fluency) and severity (Minor: issues

[8] See details in http://www.qt21.eu/mqm-definition/definition-2015-12-30.html.

that do not impede understanding of the text; Major: those that make the text difficult to understand and Critical: those that render the content unusable).[9] We also separately examine (but do not model) word prediction accuracy for each translator.

We collected translators' impressions of NITP, through a questionnaire. They rated the following on a 5-level Likert scale: *I prefer ITP to PE*; *ITP is less tiring than PE*; *As the study progressed, I took better advantage of the ITP suggestions*; *ITP helps me translate faster than PE*; *ITP helps me translate to better quality than PE* and *I would use ITP in real-life scenarios*. They also answered open questions: *Do you have any suggestions for improvement of any aspect of interactive translation's use?* and *Please provide any additional comments about your experience with interactive translation prediction*.

5 Results and analysis

We examine our data in three different ways. We begin with a quantitative analysis of our overall sample results, both averaged and broken down by individual translator. We then build mixed-effects models and examine what they can tell us about our data. Finally, we take a look at translators' impressions and feedback about the tool.

5.1 Sample results

Table 1 shows summary statistics (mean and standard deviation) for translation productivity indicators, broken down by translation condition. As Table 1 indicates, sample results for eight out of the eleven variables are favorable to ITP. Note that no Critical issues were found in any translation condition, likely due to participants being professional translators.

Exploratory graphs did not show consistent trends over time in ITP in any of the measured variables except for Mouse Clicks, which showed a steady decrease, from the first ITP session (M = 0.34, SD = 0.40), gradually to the last ITP session (M = 0.28, SD = 0.46), possibly indicating that translators change how they interact with the computer in ITP over time.

As Table 2 shows, the effect of ITP on individual translators' productivity indicators varies. All translators made more Navigation and Special Key presses and fewer Manual Deletions in ITP, and all but two (TrC and TrD) made fewer Mouse Clicks in ITP. The increase in Special Key presses is directly attributable to the use of the TAB key to accept translation suggestions in the ITP interface. Additionally, all but one translator (TrA) produced texts with more Fluency Issues in ITP, and all but one translator (TrC) produced texts with fewer Adequacy Issues in ITP.

We observe a wide range of word prediction accuracy scores (obtained by rerunning ITP as a simulation on the final translator output) for both ITP and PE, showing (as also shown in the tokens of MT origin) that the usefulness of the sugges-

[9] Quality annotation was conducted by the second author of this work.

Table 1 Summary statistics for translation productivity indicators in ITP and PE

	ITP		PE	
	Mean	SD	Mean	SD
Processing time (seconds per source token)	4.56	3.88	4.79	6.31
Manual insertions (count per source token)	2.55	2.31	3.52	3.85
Manual deletions (count per source token)	1.18	1.54	3.37	3.78
Navigation and special key presses (count per source token)	1.13	0.66	0.29	0.48
Mouse clicks (count per source token)	0.31	0.41	0.54	0.45
Tokens of MT origin (count per 100 source tokens)	61.93	30.22	59.36	33.33
MQM score (percentage)	98.52	4.01	98.25	6.02
Fluency issues (count per 1000 source tokens)	6.18	17.91	2.14	8.62
Adequacy issues (count per 1000 source tokens)	4.71	15.87	8.12	25.58
Minor issues (count per 1000 source tokens)	9.9	23.11	8.19	25.52
Major issues (count per 1000 source tokens)	0.97	6.3	2.08	12.22

tions varies by translator. In all cases, the word prediction accuracy for a translator using ITP is higher than the reference-simulated overall word prediction accuracy (59.1%). While there is not a strict correlation between the positivity of translator reactions to ITP and word prediction accuracy or Tokens of MT Origin, the three translators with the highest word prediction accuracy do agree strongly or agree that they would use ITP in real-life scenarios, while the translator with the lowest word prediction accuracy strongly disagreed. The two translators with the most PE experience would use ITP and have high word prediction accuracy scores, which may suggest that they are adept at using machine translation output in their translations.

Four translators were faster in ITP, the same number (though not the exact same set) that applied fewer Manual Insertions and made more use of MT in ITP, as measured by Tokens of MT Origin.[10] This is similar to earlier studies that have found notable between-translator variation. We discuss potential reasons for variation in Sect. 5.3.

[10] In particular, TrG's "outlying" indicators in PE are partly due to her replacing English quotation marks in the translation by guillemets: she copied the guillemets from an outside source and pasted them into CASMACAT's interface, but in the process she also pasted whitespaces and line breaks (adding to the count of Manual Insertions), some of which she then manually deleted, hence the higher temporal and technical effort indicators.

Table 2 Translators' main translation productivity indicators and impressions

		TrA	TrB	TrC	TrD	TrE	TrF	TrG	TrH
PE cert.		N	N	Y	N	N	Y	N	N
PE exp. (years)		2–5	0	2–5	2–5	5–10	5–10	<2	2–5
I prefer ITP		+	- -	+	++	+	+	-	=
I'd use ITP		+	- -	+	+	++	++	-	-
Processing	ITP	3.19	–	2.55	5.43	5.84	3.2	5.89	5.9
Time	PE	2.42	–	2.57	3.56	7.04	3.63	9.4	4.98
Manual	ITP	3.56	–	1.15	4.15	1.98	0.76	1.7	4.96
Insertions	PE	3.9	–	3.21	1.67	1.49	1.92	8.73	4.01
Manual	ITP	1.2	–	1.95	1.15	1.12	0.47	0.75	1.62
Deletions	PE	3.78	–	3.18	1.49	1.45	1.89	8.43	3.68
Nav. and	ITP	1.21	–	1.88	0.82	0.98	1.31	1.08	0.6
Special key	PE	0.49	–	0.68	0.08	0.03	0.72	0.03	0.06
Mouse	ITP	0.14	–	0.37	0.67	0.14	0.11	–	0.49
Clicks	PE	0.32	–	0.3	0.45	0.6	0.34	0.91	0.85
Tokens of	ITP	55.63	–	81.73	35.85	68.91	86.48	61.8	37.75
MT origin	PE	53.62	–	59.33	74.9	79.17	75.23	21.51	49.68
WPA	ITP	65.92	–	78.82	59.31	76.58	83.92	68.36	61.01
	PE	68.04	–	69.38	79.27	76.56	76.50	37.32	68.51
MQM Score	ITP	99.51	–	98.22	99.23	98.01	97.95	98.52	98.42
	PE	99.4	–	98.65	99.25	98.51	97.13	96.05	98.6
Fluency	ITP	2.51	–	3.08	5.47	9.57	13	6.73	2.23
	PE	3.54	–	0.49	3.22	1.91	1.43	4.66	0
Adequacy	ITP	1.90	–	8.06	2.14	5.47	4.02	3.47	7.18
	PE	3.08	–	4.29	4.47	9.49	22.96	9.66	3.55
Minor	ITP	4.04	–	9.41	7.62	14.10	15.94	9.19	7.78
	PE	6.61	–	1.81	7.69	10.39	22.96	6.84	1.40
Major	ITP	0.37	–	1.72	0	0.98	0.99	1.01	1.55
	PE	0	–	2.96	0	1.01	1.43	7.48	2.15

As in Table 1, Processing Time is measured in seconds per source token; Manual Insertions, Manual Deletions, Navigation and Special Key Presses, and Mouse Clicks are measured as counts per source token. Tokens of MT origin as measured as counts per 100 source tokens; MQM Score as a percentage, and translation issues, as count per 1000 source tokens. Word Prediction Accuracy (WPA) is a percentage. Likert responses are ranked from most negative to most positive: - -; -; =; +; ++

5.2 Mixed-effect models of translator productivity

Data was analyzed with mixed-effect models,[11] a type of regression model useful for the analysis of grouped data, that describe the effects on a response variable of one

[11] In a mixed-effects model, a linear regression model is fit, where the known values from the experimental design form the features (in one matrix for fixed effects and another for random effects), and parameters are estimated to maximize the probability of the observations.

or more explanatory variables by incorporating both fixed and random effects. Fixed effects measure population effects, while random effects control for variations in the measured variables across subjects (translators) and items (sentences).

Given that our exploratory results showed, as is common in these kinds of studies, between-subject and between-sentence variations in all variables observed, mixed-effect models were deemed appropriate to analyze the data. Additionally, our study presented a number of missing observations (see Sect. 5.3 for details), making mixed-effects models a better choice over rm-ANOVA as the former are better equipped for handling unbalanced designs caused by missing data. Additionally, the inclusion of random effects often leads to more precise estimates of the fixed effects (Fahrmeir 2013).

Translation Condition (PE/ITP) and ITP Session (2 to 8) were treated as fixed effects to address the questions of translator productivity and change over time respectively, with translators and sentences modeled as crossed random effects. To minimize Type I errors, following Barr et al. (2013), the structure of the random effects was kept maximal (all possible random effects that the design justified, and that data allowed, were included, i.e. by-subject and by-item random intercepts and slopes). Where this structure proved too complex it was progressively simplified by removing the random effects with the lowest *SD*. Untransformed Processing Time was modeled with robust linear mixed-effect models fit with robustlmm (Koller 2016). All other response variables were modeled with generalized linear mixed-effects models, fit with lme4 (Bates et al. 2015). Confidence intervals and *p* values for the fixed effects were obtained with Wald tests.

Table 3 contains the coefficients estimated by the models addressing translator productivity. Note that ITP is the reference category against which PE is compared, and as such, the point estimates next to PE represent the difference between PE and ITP. All response variables were entered in the models on a by-sentence basis. They were not normalized by sentence length, similarly to Läubli et al. (2013) because the variation introduced by sentence length was already captured by the inclusion of by-sentence random effects. Accordingly, the coefficients, and their transformations if applicable, should be interpreted on a sentence-level.

Note that, while sample results for Processing Time narrowly favor ITP, as shown in Table 1, this result is heavily influenced by TrG logging the slowest Processing Time in PE, as shown in Table 2 in that same section. The robust model presented in Table 3 downweighted the Processing Time observations for TrG in the PE condition the most of all; removing TrG's data would make Processing Time favorable to PE in both sample results ($M_{ITP} = 4.34$, $SD_{ITP} = 3.41$; $M_{PE} = 4.08$, $SD_{PE} = 5.28$) and model results, with the average sentence in PE taking -17.15 s to process (*CI* $[-26.19, -8.11]$) than the average sentence in ITP.

As Table 3 shows, ITP significantly decreases Manual Deletions and Mouse Clicks, and significantly increases Navigation and Special Key Presses and Fluency Issues. When thinking of the nature of the ITP and PE tasks, findings relative to temporal and technical effort are intuitive. Translators may automatically insert full translations in ITP without manually deleting any text: they may only need to perform Manual Deletions if they want to change a previously accepted translation suggestion or their own typed translation. In terms of Navigation and Special Key Presses, to insert a one-

Table 3 Summary parameters, standard errors and significance of models (translator productivity)

		Estimate	SE	95% C.I	p value
Processing time	ITP	108.99	15.62	[78.37, 139.61]	
	PE	− 7.7	6.3	[−17.58, 2.16]	−
Manual insertions	ITP	3.72	0.21	[3.18, 4.26]	
	PE	0.28	0.29	[− 0.36, 0.78]	0.465
Manual deletions	ITP	4.75	0.42	[3.93, 5.58]	
	PE	3.64	1.01	[1.65, 5.63]	<.001***
Nav. and special key	ITP	5.3	0.36	[4.6, 5.99]	
	PE	− 3.28	0.38	[− 4.03, − 2.53]	<.001***
Mouse clicks	ITP	1.58	0.24	[1.11,2.067]	
	PE	0.79	0.22	[0.36,1.22]	<.001***
Tokens of MT origin	ITP	3.89	0.26	[3.38, 4.4]	
	PE	− 0.27	0.46	[− 1.18, 0.63]	0.55
Pass status	ITP	2.97	0.26	[2.47, 3.48]	
	PE	− 0.09	0.51	[− 1.08, 0.90]	0.863
Fluency issues	ITP	− 2.28	0.24	[− 2.77, − 1.81]	
	PE	− 0.77	0.36	[− 1.47, − 0.07]	.03*
Adequacy issues	ITP	− 2.42	0.2	[− 2.8, − 2.04]	
	PE	0.36	0.28	[− 0.2, 0.91]	0.209
Minor issues	ITP	0.47	0.05	[0.38, 0.56]	
	PE	− 0.07	0.05	[− 0.16, 0.02]	0.14
Major issues	ITP	0.15	0.03	[0.10 , 0.20]	
	PE	0.03	0.05	[− 0.07, 0.13]	0.595

Processing Time is untransformed (seconds per sentence); Poisson family and log link function was applied to Mouse Clicks, Fluency, and Adequacy Issues; binomial family and logit link to Pass Status (a dichotomization of the MQM score); and Poisson family and square root link to all remaining variables. Observation-level random effects were added to Poisson models to model overdispersion
***$p < .001$, *$p < .05$, all two-tailed

word MT suggestion in ITP, the translator has to press TAB (a Special Key), whereas in PE, the text is already on the target side box. As for Mouse Clicks, translators usually click on the places in the target text where they are going to apply corrections to the text, and, with no initial static text to correct, translators do not need to use the mouse as much as in PE.

Regarding final translation quality, our model indicates that Fluency Issues are more than twice as frequent in ITP as in PE. It should be noted that the implementation of CASMACAT used in this study did not have a working spell checker, something that very likely contributed to the presence of fluency issues in the final texts, done both in PE and especially in ITP.[12] Specifically, the biggest contributors to Fluency issues in

[12] In addition to the lack of spell checking, a bug in tokenization for ITP may have introduced some spelling errors when the translator's spacing (for example, leaving whitespace between a number and the character "%") did not match the automatic detokenization performed by the system on the backend. This resulted in system suggestions of words with a character missing. These errors were quite rare and only reported

ITP were style (35%) and spelling, i.e., awkward language (34%), followed by minor grammar issues (30%), with the remaining 1% being major grammar issues. In PE, the biggest contributor to Fluency Issues were style issues (54%), followed by minor grammar issues (23%), spelling (15%) and major grammar issues (8%). Note that all spelling and style issues were classified as being of minor severity. Finally, to keep this finding in perspective, it is important to bear in mind that the frequency of Fluency Issues was only one of the levels on which translation quality was measured. In fact, except for one translator-session combination (TrC in the 1st ITP session) the overall MQM score stays consistently above 95%, the minimum arbitrary quality threshold.

In terms of improvement over time, none of the models could determine whether productivity indicators improved over time in ITP, with only Mouse Clicks showing a downward, but not quite significant ($p = .06$), trend. A look at the standard errors and the width of the confidence intervals of these non-significant models (not included here for reasons of space) shows that, given the potential effect sizes, larger samples would be needed to clarify the nature of the relationships between variables.

5.3 Translators' impressions

Translators' impressions of ITP were overall very positive. Out of eight translators, five agreed (TrA, TrC, TrE, TrF) or strongly agreed (TrD) that they preferred translating with ITP assistance over PE. Five translators agreed (TrA, TrC, TrD) or strongly agreed (TrE, TrF) that they would use ITP in real-life translation scenarios. Six translators agreed (TrD, TrE, TrF, TrG,TrH) or strongly agreed (TrA) that they took better advantage of ITP suggestions as the study progressed. Three translators agreed (TrG,TrH) or strongly agreed (TrD) that ITP was less tiring than PE, with one strongly disagreeing (TrB) and the rest giving neutral answers. Translators' perceptions of their own individual translation speed under ITP relative to PE showed a high number (five) of neutral responses, highlighting perhaps the difficulty of making this kind of judgment. Their answers showed differences between translators' perceived and actual quality in both conditions, with only one of the five non-neutral responses matching the annotated translation quality level.

Translators' answers to the open questions reveal a number of valuable insights into various aspects of this study. Two translators (TrB, TrH) considered the speed with which translation suggestions appeared to be a hurdle when translating. While the vast majority of translation suggestions were passed to the interface in under 100 ms, these translators may have encountered a slower translation, experienced network lag, or encountered the end of a full suggestion (end-of-sentence token generated on the backend) without realizing this, and found themselves waiting. We provide additional notes on speed in Appendix B. Four translators (TrB, TrC, TrD,TrH) pointed out the orthographic, grammar, translation, style, and discourse-level issues of the MT suggestions. Three translators (TrA,TrD,TrE) identified desirable UI features such

Footnote 12 continued
by one translator. Spelling errors (including Spanish vs. Catalan spelling differences) were also introduced naturally by translators. It is possible that translators did not catch these errors before continuing to the next sentence, perhaps due to the lack of spell checker or if they were less thorough than they would typically be in checking their translations.

as keyboard shortcut customization and search and replace options. Three translators (TrC, TrE, TrF) indicated that the varying level of MT quality from sentence to sentence made some MT suggestions for some sentences confusing, which led TrF to opt in such cases for a PE-style solution (i.e., accepting all suggestions and then post-editing the complete sentence). Three translators mentioned how some time had to elapse before making the most out of ITP:

"As the experience went on [ITP] helped me finish the tasks in a shorter time and with a higher level of confidence in the quality of my work." (TrA);
"By the end of the study I found [ITP] to be a user friendly and straightforward tool" (TrF);
"I had the distinct feeling that, on average, the suggestions were more and more spot on as I proceeded"[13] (TrD).

Two translators (TrA, TrG) noted the cognitive and translation process differences between ITP and PE, such as ITP resulting in "less time researching terminology" (TrG) and it involving "a mental process different to PE, consisting of constantly comparing ITP's suggestions to the translator's own mental translations, a process that, while seemingly complex, nevertheless sped up translation times" (TrA). Two translators (TrB, TrF) mentioned how not being able to see, in principle, the whole machine-translated text in ITP slowed the overall translation workflow, because otherwise one could decide in one look whether or not the MT output was going to be helpful.[14] Finally, two translators (TrA, TrG), expressed their worries about the translator's role and imprint in an MT-centered scenario: how in such scenarios, MT priming means "the voice of the translator is lost" (TrG), and how the user-friendliness and speed of the ITP system may generate overconfidence on the translator side and "lead to mistakes or wrong decisions if the required exigence and rigor levels are not there, on the user's side" (TrA).

Overall, translators' positive feedback towards ITP is consistent with Koehn (2009), and with Langlais et al. (2000). Only one translator (TrB) openly rejected ITP, as evidenced by him strongly disagreeing to all close-ended questions, and expressing negative views in the open questions. TrB chose, against task instructions, to ignore the ITP assistance altogether after just one ITP session, not accepting a single token the ITP system suggested afterwards, instead consistently typing his own translations, even when they matched. The translation activity data produced by TrB was deemed invalid and discarded, as any measures collected would not be representative of working in ITP, but rather of unassisted translation.[15] TrB's negative perception may have been partly due to speed reasons, as reported in his feedback; nevertheless, it seems a fairly harsh judgment after having tried ITP only once. It may be that some translators are not willing to engage with PE or ITP, possibly because they already have a working routine they are comfortable with. In this sense, the views expressed by Vasconcellos

[13] While our setup did not include adaptation to translator corrections, future work could create additional gains by doing so, as in Kothur et al. (2018); Peris and Casacuberta (2018).

[14] Showing the full sentence is a display option, which we did not examine in this work.

[15] We did compute word prediction accuracy on his translations in simulation; with a score of 52.0%, he was the only translator who had a lower word prediction accuracy than the simulated WMT test set word prediction accuracy.

and León (1985), O'Brien (2002), Rico Pérez and Torrejón (2012) and De Sutter (2011) that PE requires that the translator has a positive attitude towards MT resonate for ITP.[16]

In attempting to relate translators' feedback to their own background and quantitative results, our data can give us some useful insights into their potential relationship. While Moorkens and O'Brien (2015) reported that PE was perceived by some translators as a tedious activity, and that experienced translators were more likely to have negative views of PE than novice translators, in our study, we did not find any indication that experienced translators had negative views towards ITP, provided they also had PE experience. In fact, the more experienced translator (TrA), both in terms of length of experience (> 10 years) and translation volume in the previous 12 months (> 55k words)—who had between two and five years' PE experience—expressed, as detailed above, consistently positive views of ITP. In terms of translation productivity indicators, as shown in Table 2, TrA logged the fastest PE time and the second fastest ITP time of all translators. TrA also produced the highest quality texts in the PE condition and the highest quality texts of all translators in the ITP condition.

Regardless of their translation experience, professional translators with little or no PE experience though, may be more reluctant to engage in ITP. The two translators who expressed negative views of ITP—TrG to a minor degree, and, much more markedly, TrB—had, respectively, less than two years' PE experience or no PE experience. Finally, there is some indication that translators who have formal PE training or provide PE services frequently benefited the most from ITP. In fact, of the four translators who were faster in ITP than in PE, two have PE industry certifications (TrC [TAUS]; TrF [SDL]) and one (TrE) provides frequent PE services.

6 Conclusions

As is usual with research studies on translation processes, the empirical work discussed here presents several limitations which may have had an influence on the obtained results. Specifically, while CASMACAT is suitable for research on translation processes due to its extensive logging capabilities and it being open-source and web-based, it does not present features that are common in commercial CAT tools, such as multiple search and replace or spell checkers, to which professional translators are accustomed. This limited functionality may have contributed to slowing down translators and, most likely, to the presence of spelling issues. Additionally, cost and convenience motivated our sample size, quality assessment and language pair choices, therefore restricting the application of our findings.

With the above limitations in mind, the ITP study presented here and Daems and Macken (2019) in this special issue are, to the best of our knowledge, the first empirical studies investigating human translators' productivity in a NITP setting. Overall, our results point at ITP being a viable alternative to PE considering translators' feedback and temporal, technical and translation quality indicators. Translation requires,

[16] In hindsight, we recommend that when hiring translators for empirical studies such as the one presented here, one should highlight the importance of the 'core competence' of following PE instructions, described in Rico Pérez and Torrejón (2012) as part of a set of desired PE skills.

in many cases, long hours in front of a computer, and translating with a type of MT assistance that is overall perceived in a positive light may increase the level of job satisfaction of those translators who find it beneficial to incorporate MT to their translation workflows. We expect that the fact that sample findings are favorable to ITP in most translation productivity indicators, and that the majority of translators expressed their preference for ITP over PE, encourages further research efforts into ITP research, especially into its integration in current online or desktop-based commercial CAT tools. In future studies, it may be worthwhile to more closely analyze the impact of translator experience with PE on their success and satisfaction using technologies like ITP.

Acknowledgements This work was supported, in part, by the Human Language Technology Center of Excellence (HLTCOE) through the 2016 SCALE workshop CADET, as well as by a National Science Foundation Graduate Research Fellowship under Grant No. DGE-1232825 (to Rebecca Knowles), and by a Doctoral Scholarship and Research Fund grant from the University of Auckland (to Marina Sanchez-Torron). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

Compliance with ethical standards

Ethical approval The study received ethics approval from the University of Auckland Human Participants Ethics Committee.

Appendix A

Translators' instructions

You are about to translate on CASMACAT, a CAT tool implementing Interactive Translation Prediction (ITP). Under the ITP paradigm, the CAT tool suggests a completion of your translation as you type. To accept suggestions, word by word, press TAB. The cursor will be moved to the beginning of the next word in the translation suggestion. You can ignore the suggested translation and type yours; a new, updated suggestion based on your input will appear, which you can accept or ignore, and so on.

Other useful keyboard shortcuts (such as to move between lines and to delete entire lines) are listed in [link].

Please respect the following directions, essential to maintain the degree of control this study requires (read carefully):

(a) Work on each text in one go, without stops. Approach the translation as you would normally: you can go up and down the document and review previously translated segments and consult any translation resources if necessary.
(b) Only click on the links to access the texts when you are ready to start working on them and do not anticipate interruptions in the next 60–75 min, the approximate time it will take you to process each text.

Your aim is to produce high-quality, publishable texts. Please respect the following guidelines:

– Use as much of the machine translation output as possible.
– Do not engage in preferential changes that do not improve the quality of the text.

You do not have to follow any style guidelines. However, aim at maintaining text coherence in matters of punctuation, proper nouns, etc.

You must confirm all segments you translate, either by clicking on the TRANSLATED button, or by hitting CTRL+ENTER. Confirmed segments will have a blue vertical line on the right of the text box. If segments are not confirmed, the information necessary for this study is not stored. It is recommended to confirm all segments as you translate.

Appendix B: Speed in tool integration and implementation

In CASMACAT, the server expects to receive full sentence translation output from the machine translation server each time that a new translation or prefix completion is requested and subsequently returned to it (Alabau et al. 2014). As noted in Knowles and Koehn (2016), CPU implementations of NITP are too slow to be used in a real-life setting, and very long sentences may also be difficult to translate fast enough even with a GPU. In order to deliver translation predictions at an adequate speed, we perform translation using a GPU and implement several time-saving approaches. We use a NVIDIA GeForce GTX 1080 GPU, which can decode one token every 3.7ms, on average.

We employ the following optimizations from Knowles and Koehn (2016):

Precompute We precompute the initial translation for each sentence. We allow a long time limit for this (5 seconds) as it is done in the background when the page opens, before translators begin translating. We also limit the output to 100 tokens.

Timeout At any other point, when we are computing the predicted translation suffix for a translator-produced prefix, we only continue generating token predictions while fewer than 80 ms have elapsed. This does mean that sometimes we will be left with only a partial sentence completion, which we attempt to turn into a full sentence using patching (described below).

Patch When the prediction of the remaining tokens in the sentence stops early due to timeout, we patch together the current tokens and the end of a previous longer (complete) translation. (If none exists, we simply return the partial translation without patching.) Assuming a longer previous translation exists, we select where to patch using KL-divergence between probability distributions, as described in Knowles and Koehn (2016).

Cache We also perform caching to improve speed. As we produce a hypothesis translation, we also save the hidden states and probability distributions that were used to produce that hypothesis. That way, if the translator accepts part of the hypothesis but then diverges from it, we do not need to recompute those values, and can simply consume the new divergent

continuation of the translation before predicting new tokens. In future work, we plan to implement caching by translator and/or by document.

We aim to return each suggestion to the translator in under 100 ms, in order to avoid the user sensing a lag. In our study, 73.5% of the suggestion requests during valid ITP sessions were returned to the user in under 100 ms (99.1% in under 300 ms). Nevertheless, some users reported experiencing delays, likely due to: (1) encountering one of the slower response times or (2) accepting all tokens and reaching the end of the current prediction (after which point new suggestions will not be generated until the translator makes a change to the prefix) or (3) network lag (the server was located in the United States, and the translators, based in Europe, accessed the tool through a web interface). To mitigate the first, future work could use a faster NMT decoder adapted for ITP, or set lower thresholds. For the second, we could change the interaction between the user interface and the MT backend such that accepting a token triggers additional translation (if the suggestion produced so far has not yet reached an end-of-sentence token).

References

Alabau V, Buck C, Carl M, Casacuberta F, García-Martínez M, Germann U, González-Rubio J, Hill R, Koehn P, Leiva L, Mesa-Lao B, Ortiz-Martínez D, Saint-Amand H, Sanchis Trilles G, Tsoukala C (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, Association for Computational Linguistics, Gothenburg, Sweden, pp 25–28, http://www.aclweb.org/anthology/E14-2007

Alabau V, Carl M, Casacuberta F, Garca-Martnez M, Gonzlez-Rubio J, Mesa-Lao B, Ortiz-Martnez D, Schaeffer M, Sanchs-Trilles G (2016) Learning advanced post-editing. In: Carl M, Bangalore S, Schaeffer M (eds) New directions in empirical translation process research : exploring the CRITT TPR-DB. Springer, Berlin, pp 95–110

Alves F, Koglin A, Mesa-Lao B, Garca-Martnez M, de Lima Fonseca NB, de Melo SA, Gonalves JL, Szpak KS, Sekino K, Aquino M (2016) Analysing the impact of interactive machine translation on post-editing effort. In: Carl M, Bangalore S, Schaeffer M (eds) New directions in empirical translation process research : exploring the CRITT TPR-DB. Springer, Berlin, pp 77–94

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: ICLR, arXiv:1409.0473v6.pdf

Barr DJ, Levy R, Scheepers C, Tily HJ (2013) Random effects structure for confirmatory hypothesis testing: Keep it maximal. Journal of Memory and Language 68(3):255–278

Barrachina S, Bender O, Casacuberta F, Civera J, Cubel E, Khadivi S, Lagarda A, Ney H, Toms J, Vidal E, Vilar JM (2009) Statistical approaches to computer-assisted translation. Computational Linguistics 35(1), http://www.aclweb.org/anthology/J09-1002

Bates D, Mächler M, Bolker B, Walker S (2015) Fitting linear mixed-effects models using lme4. Journal of Statistical Software 67(1):1–48. https://doi.org/10.18637/jss.v067.i01

Bender O, Hasan S, Vilar D, Zens R, Ney H (2005) Comparison of generation strategies for interactive machine translation. In: Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT), Budapest, http://www-i6.informatik.rwth-aachen.de/publications/download/276/Bender-EAMT-2005.pdf

Daems J, Macken L (2019) Interactive adaptive SMT versus interactive adaptive NMT: a user experience evaluation. Mach Transl 33(1)

De Sutter N (2011) Mt evaluation based on post-editing: a proposal. In: Depraetere I (ed) Perspectives on translation quality. De Gruyter Mouton, Berlin and Boston, pp 125–144

Fahrmeir L (ed) (2013) Regression models, methods and applications. Springer, Berlin, New York. https://doi.org/10.1007/978-3-642-34333-9

Foster G, Langlais P, Lapalme G (2002) User-friendly text prediction for translators. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Philadelphia, pp 148–155, http://acl.ldc.upenn.edu/W/W02/W02-1020.pdf

Green S, Heer J, Manning CD (2013) The efficacy of human post-editing for language translation. In: 2013 IGCHI Conference on Human Factors in Computing Systems, pp 439–448

Green S, Chuang J, Heer J, Manning CD (2014) 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, pp 177–187

Knowles R, Koehn P (2016) Neural interactive translation prediction. In: Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)

Koehn P (2005) Europarl: A parallel corpus for statistical machine translation. MT Summit 5:79–86

Koehn P (2009) A process study of computer-aided translation. Machine Translation 23(4):241–263, http://www.researchgate.net/publication/220419195_A_process_study_of_computer-aided_translation/file/60b7d5149f75e6f7d0.pdf

Koehn P, Tsoukala C, Saint-Amand H (2014) Refinements to interactive translation prediction based on search graphs. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Baltimore, Maryland, pp 574–578, http://www.aclweb.org/anthology/P14-2094

Koller M (2016) robustlmm: An R package for robust estimation of linear mixed-effects models. Journal of Statistical Software 75(6):1–24. https://doi.org/10.18637/jss.v075.i06

Kothur SSR, Knowles R, Koehn P (2018) Document-level adaptation for neural machine translation. In: Proceedings of the Second Workshop on Neural Machine Translation and Generation, Association for Computational Linguistics, Melbourne

Langlais P, Foster G, Lapalme G (2000) Transtype: a computer-aided translation typing system. In: Proceedings of the ANLP-NAACL 2000 Workshop on Embedded Machine Translation Systems, http://acl.ldc.upenn.edu/W/W00/W00-0507.pdf

Läubli S, Fishel M, Massey G, Ehrensberger-dow M, Volk M (2013) Assessing post-editing efficiency in a realistic translation environment. In: Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice, pp 83–91

Lin D (1996) On the structural complexity of natural language sentences. In: 16th International Conference on Computational Linguistics (COLING-96), pp 729–733

Macklovitch E (2006) Transtype2: The last word. In: Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 06), pp 167–172

Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ, McClosky D (2014) The stanford corenlp natural language processing toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp 55–60

Massardo I, van der Meer J, O'Brien S, Hollowood F, Aranberri N, Drescher K (2011) Taus/cngl machine translation post-editing guidelines

Mishra A, Bhattacharyya P, Carl M (2013) Automatically predicting sentence translation difficulty. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp 346–351

Moorkens J, O'Brien S (2015) Post-editing evaluations: Trade-offs between novice and professional participants. In: Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT 2015), pp 75–81

O'Brien S (2002) Teaching post-editing: A proposal for course content. In: 6th EAMT Workshop Teaching Machine Translation, pp 99–106

Peris Á, Casacuberta F (2018) Online learning for effort reduction in interactive neural machine translation. CoRR abs/1802.03594, arXiv:1802.03594

Rico Pérez C, Torrejón E (2012) Skills and profile of the new role of the translator as mt post-editor. Revista Tradumica: tecnologies de la traducci 10:166–178

Sanchis-Trilles G, Alabau V, Buck C, Carl M, Casacuberta F, Garca-Martnez M, Germann U, Gonzlez-Rubio J, Hill RL, Koehn P, Leiva LA, Mesa-Lao B, Ortz-Martnez D, Saint-Amand H, Tsoukala C, Vidal E (2014) Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. Mach Transl 28(3):217–235

Sennrich R, Haddow B, Birch A (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), Association for Computational Linguistics, Berlin, Germany, pp 1715–1725, http://www.aclweb.org/anthology/P16-1162

Sennrich R, Firat O, Cho K, Birch A, Haddow B, Hitschler J, Junczys-Dowmunt M, Läubli S, Miceli Barone AV, Mokry J, Nadejde M (2017) Nematus: a toolkit for neural machine translation. In: Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Valencia, Spain, pp 65–68, http://aclweb.org/anthology/E17-3017

Underwood N, Mesa-Lao B, Martinez MG, Carl M, Alabau V, Gonzalez-Rubio J, A Leiva L, Sanchis-Trilles G, Ortz-Martnez D, Casacuberta F (2014) Evaluating the effects of interactivity in a post-editing workbench. Proceedings of the Ninth International Conference on Language Resources and Evaluation (lrec'14) pp 553–559

Vasconcellos M, León M (1985) Spanam and engspan: machine translation at the pan american health organization. Computational Linguistics 11(2–3):122–136

Wuebker J, Green S, DeNero J, Hasan S, Luong MT (2016) Models and inference for prefix-constrained machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, pp 66–75, http://www.aclweb.org/anthology/P16-1007

Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701