

Translator2Vec: Understanding and Representing Human Post-Editors

António Góis

Unbabel
Lisbon, Portugal

antonio.gois@unbabel.com

André F. T. Martins

Unbabel & Instituto de Telecomunicações
Lisbon, Portugal

andre.martins@unbabel.com

Abstract

The combination of machines and humans for translation is effective, with many studies showing productivity gains when humans post-edit machine-translated output instead of translating from scratch. To take full advantage of this combination, we need a fine-grained understanding of how human translators work, and which post-editing styles are more effective than others. In this paper, we release and analyze a new dataset with document-level post-editing action sequences, including edit operations from keystrokes, mouse actions, and waiting times. Our dataset comprises 66,268 full document sessions post-edited by 332 humans, the largest of the kind released to date. We show that action sequences are informative enough to identify post-editors accurately, compared to baselines that only look at the initial and final text. We build on this to learn and visualize continuous representations of post-editors, and we show that these representations improve the downstream task of predicting post-editing time.

1 Introduction

Computer-aided translation platforms for interactive translation and post-editing are now commonly used in professional translation services (Alabau et al., 2014; Federico et al., 2014; Green et al., 2014; Denkowski, 2015; Hokamp, 2018; Sin-wai, 2014; Kenny, 2011). With the increasing quality of machine translation (Bahdanau

et al., 2014; Gehring et al., 2017; Vaswani et al., 2017), the translation industry is going through a transformation, progressively shifting gears from “computer-aided” (where MT is used as an instrument to help professional translators) towards **human-aided translation**, where there is a human in the loop who only intervenes when needed to ensure final quality, and whose productivity is to be optimized. A deep, data-driven understanding of the **human post-editing process** is key to achieve the best trade-offs in translation efficiency and quality. What makes a “good” post-editor? What kind of behaviour shall an interface promote?

There is a string of prior work that relates the difficulty of translating text with the cognitive load of human translators and post-editors, based on indicators such as editing times, pauses, keystroke logs, and eye tracking (O’Brien, 2006; Doherty et al., 2010; Lacruz et al., 2012; Balling and Carl, 2014, see also §6). Most of these studies, however, have been performed in controlled environments on a very small scale, with a limited number of professional translators and only a few sessions. A direct use of human activity data for understanding and representing human post-editors, towards improving their productivity, is still missing, arguably due to the lack of large-scale data. Understanding how human post-editors work could open the door to the design of better interfaces, smarter allocation of human translators to content, and automatic post-editing.

In this paper, we study the behaviour of human post-editors “in the wild” by automatically examining tens of thousands of post-editing sessions at a document level. We show that these detailed editor activities (which we call **action sequences**, §2) encode useful additional information

© 2019 The authors. This article is licensed under a Creative Commons 4.0 licence, no derivative works, attribution, CC-BY-ND.

besides just the initial machine-translated text and the final post-edited text. This is aligned to recent findings in other domains: Yang et al. (2017) and Faruqui et al. (2018) have recently shown that Wikipedia page edits can represent interesting linguistic phenomena in language modeling and discourse. While prior work analyzed the cognitive behaviour of post-editors and their productivity by collecting a few statistics, we take a step forward in this paper, using state-of-the-art machine learning techniques to **represent editors in a vector space** (§4). These representations are obtained by training a model to **identify** the editor based on his action sequences (§3). This model achieves high accuracy in predicting the editor’s identity, and the learned representations exhibit interesting correlations with the editors’ behaviour and their productivity, being effective when plugged as features for **predicting the post-editing time** (§5).

Overall, we use our action sequence dataset to address the following research questions:

1. **Editor identification** (§3): are the post-editors’ activities (their action sequences) informative enough to allow discriminating their identities from one another (compared to just using the initial machine-translated text and the final post-edited one)?
2. **Editor representation** (§4): can the post-editors’ activities be used to learn meaningful vector representations, such that similar editors are clustered together? Can we interpret these embeddings to understand which activity patterns characterize “good” editors (in terms of translation quality and speed)?
3. **Downstream tasks** (§5): do the learned editor vector representations provide useful information for downstream tasks, such as predicting the time to translate a document, compared to pure text-based approaches that do not use them?

We base our study on editor-labeled action sequences for two language pairs, English-French and English-German, which we make available for future research. In both cases, we obtain positive answers to the three questions above.

2 Post-Editor Action Sequences

A crucial part of our work is in converting raw keystroke sequences and timestamps into **action**

Action	Symbol	Appended Info
Replace	R	new word
Insert	I	new word
Delete	D	old word
Insert Block	BI	new block of words
Delete Block	BD	old block of words
Jump Forward	JF	# words
Jump Back	JB	# words
Jump Sentence Forward	JSF	# sentences
Jump Sentence Back	JSB	# sentences
Mouse Clicks	MC	# mouse clicks
Mouse Selections	MS	# mouse selections
Wait	W	time (seconds)
Stop	S	–

Table 1: Text-editing and non-editing actions.

sequences—sequences of symbols in a finite alphabet that describe word edit operations (insertions, deletions, and replacements), batch operations (cutting and pasting text), mouse clicks or selections, jump movements, and pauses.

Each action sequence corresponds to a single post-editing session, in which a human post-edits a document. The starting point is a set of source documents (customer service email messages), which are sent for translation to Unbabel’s online translation service. The documents are split into sentences and translated by a domain-adapted neural machine translation system based on Marian (Junczys-Dowmunt et al., 2018). Finally, each document is assigned to a human post-editor to correct eventual translation mistakes.¹ These post-editing sessions are logged, and all the keystroke and mouse operation events are saved, along with timestamps. A preprocessing script converts these raw keystrokes into word-level action sequences, as we next describe, and a unique identifier is appended that represents the human editor.

The preprocessing for converting the raw character-level keystroke data into word-level actions is as follows. We begin with a sequence of all intermediate states of a document between the machine-translated and the post-edited text, containing changes caused by each keystroke. We track the position of the word currently being edited and store one action summarizing the change in that word. A single keystroke may also

¹The human post-editors are native or proficient speakers of both source and target languages, although not necessarily professional translators. They are evaluated on language skills and subject to periodic evaluations by Unbabel. Editors have access to whole documents when translating, and they are given content-specific guidelines, including style, register, etc.

Source	Hey there, Some agents do speak Spanish, otherwise our system will translate :) Best, <Name>										
MT	Bonjour, Certains agents parlent espagnol, sinon notre système <i>se traduira par</i> :) Cordialement, <Name>										
PE	Bonjour, Certains agents parlent espagnol, sinon notre système traduit :) Cordialement, <Name>										
Actions	W:23	JSF:1	JF:8	D:se	W:2	MC:1	MS:1	JF:1	D:par	W:7	
	MC:1	MS:1	JB:1	R:traduit	W:2	MS:1	S:-				

Table 2: Example of a document and corresponding action sequence. We mark in *red* the MT words that have been corrected and in *blue* their replacement. The actions used here were W (wait), JSF (jump sentence forward), JF (jump forward), D (delete), MC (mouse clicks), MS (mouse selections), JB (jump back), R (replace) and S (stop).

cause simultaneous changes to several words (e.g. when pasting text or deleting a selected block), and we reserve separate actions for these. Overall, five **text-editing actions** are considered: inserting (I), deleting (D), and replacing (R) a single word, and inserting (BI) and deleting (BD) a block of words. Each action is appended with the corresponding word or block of words, as shown in Table 1.

Other actions, dubbed **non-editing actions**, do not change the text directly. Jump-forward (JF) and jump-backward operations (JB) count the distance in words between two consecutive edits. Another pair of actions informs when a new sentence is edited: a sentence jump (JSF/JSB) indicates that we moved a certain number of sentences forth/back since the previous edit. Mouse clicks (MC) and mouse selections (MS) count their occurrences between two consecutive edits. Wait (W) counts the seconds between the beginning of two consecutive edits. Finally, stop (S) marks the end of the post-editing session.

Since we do not want to rely on lexical information to identify the human post-editors, only the 50 most frequent words were kept (most containing punctuation symbols and stop-words), with the remaining ones converted to a special unknown symbol (UNK). Moreover, the first waiting time is split in two: the time until the first keystroke occurs and, in case the first keystroke is not part of the first action (e.g. a mouse click), a second waiting time until the first action begins.

Table 2 shows an example of a small document, along with the editor’s action sequence. The editor began on sentence 2 (“Certains agents...”) and the word on position 9, since there was a jump for-

ward of 1 sentence and 8 words. After deleting “se”, position 9 became “traduira”. Since the editor opted to delete “par” (using a mouse selection) before changing the verb, there is a jump forward of 1 word to position 10. Then we have a jump back of 1 before changing the verb to “traduit”.

Datasets. We introduce two datasets for this task, one for English-French (En-Fr) and another for English-German (En-De). For each dataset, we provide the action sequences for full documents, along with an editor identifier. To ensure reproducibility of our results, we release both datasets as part of this paper, available in https://github.com/Unbabel/translator2vec/releases/download/v1.0/keystrokes_dataset.zip. For anonymization purposes, we convert all editor names and the 50 tokens in the word vocabulary to numeric identifiers. Statistics of the dataset are shown in Table 3: it is the largest ever released dataset with post-editing action sequences, and the only one we are aware of with document-level information.² Each document corresponds to a customer service email with an average of 116.6 tokens per document. Each sentence has an average length of 9.4 tokens.

²The closest comparable dataset was released by Specia et al. (2017) in the scope of the QT21 project, containing 176,476 sentences spanning multiple language pairs (about 4 times less), with raw keystroke sequences being available by request. In contrast to ours, their units are sentences and not full documents, which precludes studying how human post-editors jump between sentences when translating a document.

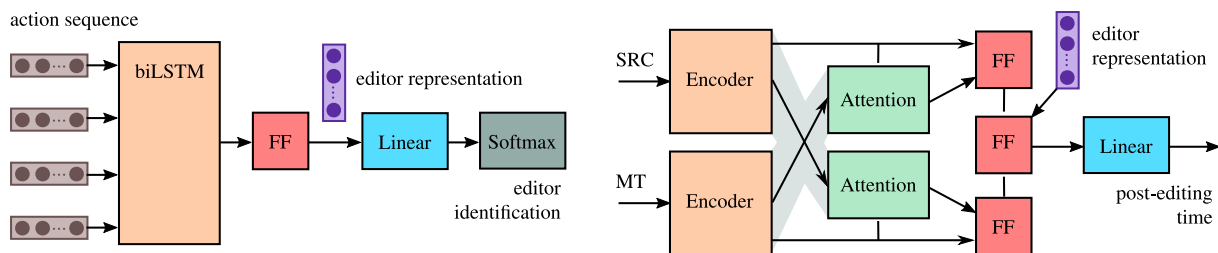


Figure 1: Left: our **Action Seq** model for editor identification. Right: our model for post-editing time prediction.

		# docs	# sents	# words
En-Fr	train	17,464	154,026	1,895,389
	dev	5,514	52,366	659,675
	test	9,441	86,111	1,072,807
En-De	train	17,403	169,478	2,053,407
	dev	6,722	66,521	826,791
	test	9,724	98,920	1,221,319
Total		66,268	627,422	7,729,388

Table 3: Number of documents, sentences, and words in English source text per dataset. There are 149 unique editors across all En-Fr datasets, and 183 in En-De.

3 Editor Identification

We now make use of the dataset just described to answer the three research questions stated at the end of §1, starting with **editor identification**.

3.1 Data Preparation

For this experiment, we took the action sequence dataset described in §2 and selected a small number of human translators for both language pairs who post-edited a number of documents above a threshold: this yielded 6 editors for En-Fr and 7 editors for En-De. To ensure balanced datasets, we filtered them to contain the same number of samples per selected editor. This filtering yielded a total of 998/58/58 training/dev/test documents per editor for En-Fr, and 641/128/72 for En-De.

A random baseline for this dataset would obtain an editor identification accuracy of $1/6 = 16.7\%$ for En-Fr and $1/7 = 14.3\%$ for En-De.

3.2 A Model for Editor Identification

Let $\langle x_1, \dots, x_L \rangle$ be an action sequence produced by a post-editor y . To identify the editor of a task, we build a model $P(y | x_1, \dots, x_L)$ using a neural network as we next describe (shown in Figure 1). Each action x_i is first associated to a one-hot vector. All numeric actions are grouped

into bins—e.g. waiting times of 200 seconds and higher all correspond to the same one-hot representation. Bins were defined manually, providing higher granularity to small values than to larger ones.³ Each one-hot is then mapped to a learnable embedding, and the sequence of embeddings is fed to a 2-layer bidirectional LSTM (biLSTM; Hochreiter and Schmidhuber (1997); Graves and Schmidhuber (2005)), resulting in two final states $\vec{h}, \overleftarrow{h}$. Then we concatenate both, apply dropout (Srivastava et al., 2014) and feed them to a feed-forward layer with a ReLU activation (Glorot et al., 2011) to form a vector h . This vector is taken as the representation of the action sequence. Finally, we define $P(y | x_1, \dots, x_L) = \text{softmax}(Wh + b)$.

We call this model **Action Seq**, since it exploits information from the action sequences.

3.3 Baselines

To assess how much information action sequences provide about human editors beyond the initial (machine translation) and final (post-edited) text, we implemented various baselines which do **not** use fine-grained information from the action sequences. All use pre-trained text embeddings from FastText (Joulin et al., 2017), and they are all tuned for dropout and learning rate:

- One using the machine-translated text only (**MT**). Since this text has not been touched by the human post-editor, we expect this system to perform similarly to the random baseline. The goal of this baseline is to control whether there is a bias in the content each editor receives that could discriminate her identity. It uses word embeddings as input to a biLSTM, followed by feed-forward and softmax layers.

³We used $\{0, \dots, 5, 7, 10, 15, 20, 30, 50, 75, 100, 150, 200+\}$ for wait and jump events (in seconds and word positions, respectively); and $\{0, \dots, 5, 7, 10+\}$ for sentence jumps and mouse events (in sentence positions and clicks).

- Another one using the post-edited text only (**PE**). This is used to control for the linguistic style of the post-editor. We expect this to be a weak baseline, since although there are positive results on translator stylometry (El-Fiqi et al., 2019), the task of post-editing provides less opportunity to leave a fingerprint than if writing a translation from scratch. The architecture is the same as in the **MT** baseline.
- A baseline combining both MT and PE using a dual encoder architecture (**MT + PE**), inspired by models from dialogue response (Lowe et al., 2015; Lu et al., 2017). This baseline is stronger than the previous two, since it is able to look at the *differences* between the initial and final text produced by the post-editor, although it ignores the process by which these differences have been generated. Two separate biLSTMs encode the two sequences of word embeddings, the final encoded states are concatenated and fed to a feed-forward and a softmax layer to provide the editors’ probabilities.
- Finally, a stronger baseline (**MT + PE + Att**) that is able to “align” the MT and PE, by augmenting the dual encoder above with an attention mechanism, inspired by work in natural language inference (Rocktäschel et al., 2016). The model resembles the one in Figure 1 (right), with a softmax output layer and without the editor representation layer. Two separate biLSTMs are used to encode the machine-translated and the post-edited text. The final state of the MT is used to compute attention over the PE, then this attention-weighted PE is concatenated with MT’s final state and passed through a feed-forward layer. Symmetrically we obtain a representation from PE’s final state and an attention-weighted MT. Finally both vectors are concatenated and turned into editors’ probabilities through another feed-forward layer.

Additionally, we prepare another baseline (**Delta**) as a tuple with meta information containing statistics about the difference between the initial and final text (still not depending on the action sequences). This tuple contains the following 5 elements: a count of sentences in the document, minimum edit distance between MT and PE, count of words in the original document, in MT and in PE. Each of these elements is binned and mapped to a learnable embedding. The 5 embeddings are

	En-De (%)	En-Fr (%)
Delta	16.15	26.09
MT	18.21	16.44
PE	27.38	30.00
MT + PE	26.63	31.78
MT + PE + Att	30.12	35.06
Action Seq	84.37	67.07

Table 4: Results of model and baselines for editor identification. Reported are average test set accuracies of 5 runs, with 7 editors for En-De and 6 editors for En-Fr.

	En-De (%)	En-Fr (%)
Action Seq	83.31	73.16
w/out editing actions	80.60	69.37
w/out mouse info	75.49	66.38
w/out waiting time	80.42	70.92
w/out 1st waiting time	78.60	71.15
only editing actions	60.20	59.08
only mouse info	56.43	55.06
only waiting time	53.53	44.02
only 1st waiting time	24.22	23.11

Table 5: Ablations studies for editor identification. Reported are average development set accuracies of 5 runs, with 7 editors for En-De and 6 editors for En-Fr.

concatenated into a vector e , followed by a feed-forward layer and a softmax activation.

3.4 Editor Identification Accuracy

Table 4 compares our system with the baselines above. Among the baselines, we observe a gradual improvement as models have access to more information. The fact that the MT baseline performs closely to the random baseline is reassuring, showing that there is no bias in the type of text that each editor receives. As expected, the dual encoder model with attention, being able to attend to each word of the MT and post-edited text, is the one which performs the best, surpassing the random baseline by a large margin. However, none of these baselines have a satisfactory performance on the editor identification task.

By contrast, the accuracies achieved by our proposed model (**Action Seq**) are striking: 84.37% in En-De and 67.07% in En-Fr, way above the closest baselines. This large gap confirms our hypothesis that **the editing process itself contains information which is much richer than the initial and final text only**.

Ablation studies. To understand the importance of each action type in predicting the editor’s identity, we conduct a series of ablation studies and report development set accuracies in Table 5. These experiments involve removing mouse information, time information, initial waiting time or editing actions. Also, we try keeping only each of the previous four. We find that all action types contribute to the global accuracy, although to different extents. Also, some action types achieve high performance on their own. Somewhat surprisingly, mouse information alone achieves remarkably high accuracy. Although waiting times also perform well on their own, removing them has little impact on the final score.

4 Editor Representation

The previous section has shown how the action sequences are very effective for identifying editors. As a by-product, the **Action Seq** model used for that task produced an internal vector h that represents the full post-editing session. This suggests a strategy for obtaining **editor representations**: simply *average* all such vectors from each editor. One way of looking at this is regarding editor identification as an auxiliary task that assists us in finding good editor representations. This draws inspiration from previous work, such as Mikolov et al. (2013), as well as its applications to recommendation systems (Grbovic et al., 2015, 2016). In the last two works, an auxiliary task also helps to provide a latent representation of an object of interest.

Visualization of translation sessions. To visualize the vectors h produced during our auxiliary task, we use Parametric t-SNE (Maaten, 2009) for dimensionality reduction. Unlike the original t-SNE (Maaten and Hinton, 2008), the parametric version allows to reapply a learned dimensionality reduction to new data. This way it is possible to infer a 2D structure using the training data, and check how well it fits the test data.

In Figure 2 we show a projection of vectors h for both language pairs, using a t-SNE model learned on the training set vectors; each color corresponds to a different editor. In the training set (used to train both the editor identification model and the Parametric t-SNE) there is one clear cluster for each editor, in both languages. Using test set data, new tasks also form clusters which are closely related to the editors’ identity. Some clusters are isolated while others get mixed near their

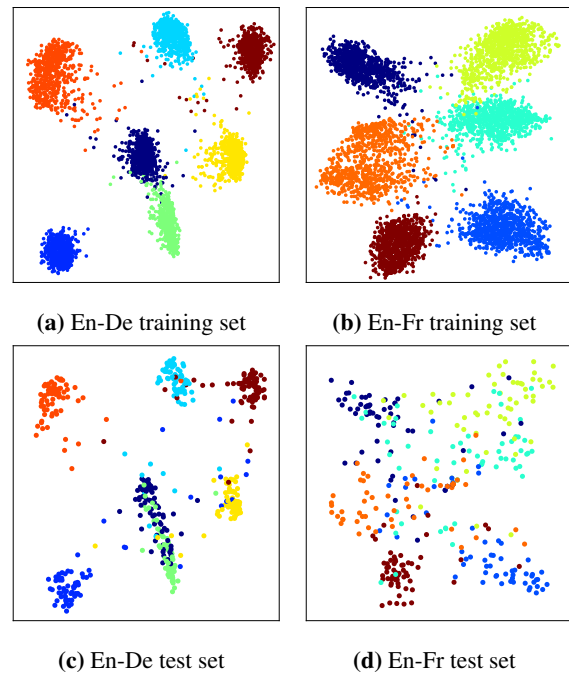


Figure 2: Embeddings of each translation session in the editor identification train and test sets, with editors identified by different colors. For each language, the dimensionality reduction was learned by training parametric t-SNE (Maaten, 2009) on the train data, and then applying it to both train and test data. En-De contains 7 editors, each with 641 train and 72 test samples per editor. En-Fr contains 6 editors, each with 998 train and 58 test samples per editor.

borders, possibly meaning that some editors behave in a more distinctive manner than others.

Visualization of editor representations. To represent an editor with a single vector, we average the h ’s of all tasks of a given editor to obtain his representation. Figure 3 contains representations for En-Fr editors (similar results have been achieved for En-De editors), using the exact same model as in Figure 2b to produce session embeddings, and the same t-SNE model for visualization. To reduce noise we discard editors with less than 10 samples, keeping 117 out of 149 editors. In Figure 3 we show percentiles for 3 editor features, using one point per editor and setting color to represent a different feature in each panel. In Figure 3a, color represents percentiles of average initial waiting time, and in Figure 3b, percentiles of counts of jump-backs per MT token. We can observe that the model learned to map high waiting times to the left and high counts of jump-backs to the right. In Figure 3c we have mouse activity per user (percentiles of counts of mouse clicks and selections). Here we can see a distribution very similar to that of count of jump-backs.

	Mouse and JB (%)	1st WT and JB (%)
En-Fr	80.75	-39.65
En-De	59.62	-31.11

Table 6: Pearson correlation between two pairs of variables: mouse actions / jump backs and first waiting time / jump backs.

We hypothesize that there are two types of human editors: those who first read the full document and then post-edit it left to right; and those who read as they type, and go back and forth. To check these hypothesis, we measure the Pearson correlation between two pairs of variables in Table 6. Indeed, there is a slight negative correlation between the average initial pause and the count of jump backs per word. This matches intuition, since a person who waited longer before beginning a task will probably have a clearer idea of what needs to be done in the first place. We also present the correlation between the count of mouse events (clicks and selections) and count of jump backs, which we observe to be very high. This may be due to the need to move between distant positions of the document, which is more commonly done with the mouse than with the keyboard.

5 Prediction of Post-Editing Time

Finally, we design a downstream task with the goal of assessing the information contained in each translator’s vector h and observing its applicability in a real-world setting. The task consists in predicting the post-editing time of a given job, which has been used as a quality estimation task in previous work (Cohn and Specia, 2013; Specia, 2011). As a baseline, we use the previously described dual encoder with attention (Figure 1, right). The inputs are the word embeddings of the original document and of the machine translation. In the output layer, instead of predicting each editor’s logit, we predict the logarithm of the post-editing time per source word, following Cohn and Specia (2013). We use mean squared error as the loss. For our proposed model, we augment this baseline by providing a “dynamic” representation of the human post-editor as described below.

Dynamic editor representations. In order to obtain an editor’s embedding in a real-time setting we do the following: For each new translation session, we store its corresponding embedding, keeping a maximum of 10 previous translations per ed-

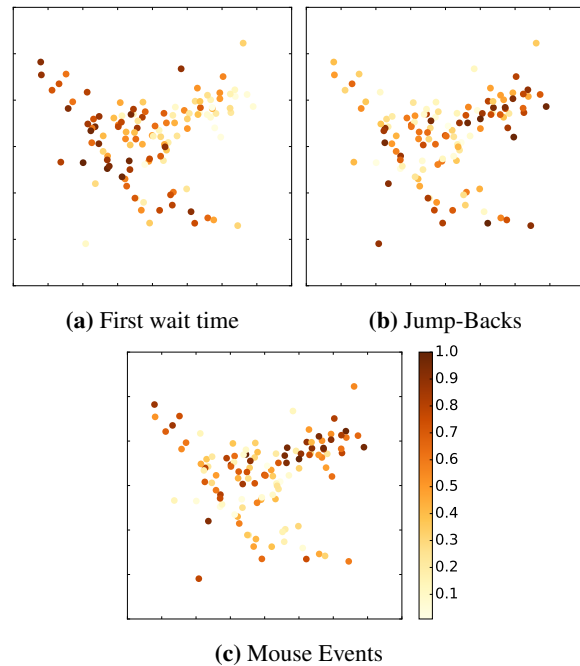


Figure 3: Embeddings of each En-Fr editor, mapped using the same parametric t-SNE as in Figure 2. In 3a we have average pause before beginning for each editor, in percentile. In 3b we have the count of jump-backs per MT token of each editor, also in percentile. In 3c we have percentiles of counts of mouse clicks and selections per editor.

itor. Whenever an editor’s embedding is required, we compute the average of his stored translations into a single vector. This allows updating the editors’ representations incrementally in a dynamic fashion, coping with the fact that editors change their behaviour over time as they learn to use the translation interface.

To introduce a translator vector h into the previously described baseline, we increase the input size of the feed-forward layer which receives both encoders’ outputs, and we introduce h in this step by concatenating it to the encoders’ outputs.

Results. Both models are evaluated using Pearson correlation between the predicted and real log-times. Results in Table 7 confirm our hypothesis that **editor representations can be very effective for predicting human post-editing time**, with consistent gains in Pearson correlation (+30.11% in En-Fr and +15.05% in En-De) over the baseline that does not use any editor information. Our approach also allows for initializing and updating editor embeddings dynamically, i.e. without having to retrain the time-prediction model.⁴

⁴This experiment also reveals that previous work on translation quality estimation (Specia et al., 2018) using time predictions can have biased results if different types of translators

		Using source text and MT (%)	Adding dynamic editor embedding (%)
En-Fr	dev	19.53	42.98
	test	17.58	47.69
En-De	dev	27.62	47.40
	test	23.67	38.72

Table 7: Pearson correlation between real and predicted logarithm of time per word in source text.

6 Related Work

There is a long string of work studying the cognitive effort in post-editing machine translation. One of the earliest instances is O’Brien (2006), who investigates the relationship between pauses and cognitive effort in post-editing. This correlation has also been studied by examination of keystroke logs (Lacruz et al., 2012; Lacruz and Shreve, 2014). Our results further confirm this, and also identify other characteristics as a fingerprint of the editors: mouse information and jumps.

More recently, Moorkens and O’Brien (2015) compare novice and professional post-editors in terms of their suitability as research participants when testing new features of post-editing environments. They conclude that professionals are more efficient but less flexible to interface changes, which confirms the existence of several editor profiles, not necessarily ones better than the others.

Other small-scale studies identify editor behaviour during translation. Asadi and Séguinot (2005) distinguish between translators who plan ahead and those who type as they think. Daems and Macken (2019) identify personal preferences between usage of mouse vs. keyboard. De Almeida (2013) studies differences and similarities in editor behaviour for two language pairs, regarding types of edits, keyboard vs. mouse usage and Web searches.

Carl et al. (2011) have shown that human translators are more productive and accurate when post-editing MT output than when translating from scratch. This has recently been confirmed by Toral et al. (2018), who have shown further gains with neural MT compared to phrase-based MT. Koponen et al. (2012) show HTER (Snover et al., 2006) is limited to measure cognitive effort, and suggest post-editing time instead. On the other hand, Herbig et al. (2019) measure cognitive effort subjectively by directly inquiring translators, and then

edit different documents. Our editor representations can be potentially useful for removing this bias.

use a combination of features to predict this cognitive effort – such task could potentially be improved by including translator representations as an additional feature. Blain et al. (2011) take a more qualitative approach to understanding post-editing by introducing a measure based on post-editing actions. Specia (2011) attempts to predict the post-editing time using quality estimation, and Koehn and Germann (2014); Sanchez-Torron and Koehn (2016) study the impact of machine translation quality in post-editor productivity. Tatsumi et al. (2012) study the effect of crowd-sourced post-editing of machine translation output, finding that larger pools of non-experts can frequently produce accurate translations as quickly as experts. Aziz et al. (2012) developed a tool for post-editing and assessing machine translation which records data such as editing time, keystrokes, and translator assessments. A similar tool has been developed by Denkowski and Lavie (2012); Denkowski et al. (2014b), which is able to learn from post-editing with model adaptation (Denkowski et al., 2014a). Our encouraging results on time prediction using editor representations suggests that these representations may also be useful for learning personalized translation models.

Yin et al. (2019) learn representations of single edits, and include a downstream task: applying these edits to unseen sentences. Wikipedia edits have been studied by Yang et al. (2017) and Faruqui et al. (2018). The latter study what can be learned about language by observing the editing process that cannot be readily learned by observing only raw text. Likewise, we study what can be learned about the translation process by observing how humans type, which cannot be readily learned by observing only the initial and final text.

Our work makes a bridge between the earliest studies on the cognitive effort of human post-editors and modern representation learning techniques, towards embedding human translators on a vector space. We draw inspiration on techniques for learning distributed word representations (Mikolov et al., 2013; Pennington et al., 2014), which have also been extended for learning user representations for recommendation systems (Grbovic et al., 2015, 2016). These techniques usually obtain high-quality embeddings by tuning the system for an auxiliary task, such as predicting a word given its context. In our case, we take **editor identification** as the auxiliary task,

given a sequence of keystrokes as input. A related problem (but with a completely different goal) is the use of keystroke dynamics for user authentication (Monrose and Rubin, 2000; Banerjee and Woodard, 2012; Kim and Kang, 2018). Unlike this literature, our paper is focused on post-editing of machine-translated text. This is more similar to El-Fiqi et al. (2019), who focus on identifying the translator of a book from his translation style. However, we are not interested in the problem of editor identification per se, but only as a means to obtain good representations.

7 Conclusions

We introduced and analyzed the largest public dataset so far containing post-editing information retrieved from raw keystrokes. We provided strong evidence that these intermediate steps contain precious information unavailable in the initial plus final translated document, by formulating and providing answers to three research questions: (i) that action sequences can be used to perform accurate editor identification; (ii) that they can be used to learn human post-editor vector representations that cluster together similar editors; and (iii) that these representations help downstream tasks, such as predicting post-editing time. In sum, we showed that fine-grained post-editing information is a rich and untapped source of information, and we hope that the dataset we release can foster further research in this area.

Acknowledgments

We would like to thank Carla Parra, Alon Lavie, Ricardo Rei, António Lopes, and the anonymous reviewers for their insightful comments. This work was partially supported by the EU/FEDER programme under PT2020 (contracts 027767 and 038510) and by the European Research Council (ERC StG DeepSPIN 758969).

References

Alabau, Vicent, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis 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.

Asadi, Paula and Candace Séguinot. 2005. Shortcuts, strategies and general patterns in a process study of nine professionals. *Meta: Journal des traducteurs/Meta: Translators' Journal*, 50(2):522–547.

Aziz, Wilker, Sheila Castilho, and Lucia Specia. 2012. Pet: a tool for post-editing and assessing machine translation. In *LREC*, pages 3982–3987.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Balling, Laura Winther and Michael Carl. 2014. *Post-editing of machine translation: Processes and applications*. Cambridge Scholars Publishing.

Banerjee, Salil P and Damon L Woodard. 2012. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139.

Blain, Frédéric, Jean Senellart, Holger Schwenk, Mirko Plitt, and Johann Roturier. 2011. Qualitative analysis of post-editing for high quality machine translation. *MT Summit XIII: the Thirteenth Machine Translation Summit [organized by the] Asia-Pacific Association for Machine Translation (AAMT)*, pages 164–171.

Carl, Michael, Barbara Dragsted, Jakob Elming, Daniel Hardt, and Arnt Lykke Jakobsen. 2011. The process of post-editing: a pilot study. *Copenhagen Studies in Language*, 41:131–142.

Cohn, Trevor and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 32–42.

Daems, Joke and Lieve Macken. 2019. Interactive adaptive smt versus interactive adaptive nmt: a user experience evaluation. *Machine Translation*, pages 1–18.

De Almeida, Giselle. 2013. *Translating the post-editor: an investigation of post-editing changes and correlations with professional experience across two Romance languages*. Ph.D. thesis, Dublin City University.

- Denkowski, Michael. 2015. *Machine translation for human translators*. Ph.D. thesis, Ph. D. thesis, Carnegie Mellon University.
- Denkowski, Michael, Chris Dyer, and Alon Lavie. 2014a. Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 395–404.
- Denkowski, Michael and Alon Lavie. 2012. Transcenter: Web-based translation research suite. In *AMTA 2012 Workshop on Post-Editing Technology and Practice Demo Session*, page 2012.
- Denkowski, Michael, Alon Lavie, Isabel Lacruz, and Chris Dyer. 2014b. Real time adaptive machine translation for post-editing with cdec and transcenter. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 72–77.
- Doherty, Stephen, Sharon OBrien, and Michael Carl. 2010. Eye tracking as an mt evaluation technique. *Machine translation*, 24(1):1–13.
- El-Fiqi, Heba, Eleni Petraki, and Hussein A. Abbass. 2019. [Network motifs for translator stylometry identification](#). *PLoS ONE*, 14(2):e0211809.
- Faruqui, Manaal, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. Wikiatomicedits: A multilingual corpus of wikipedia edits for modeling language and discourse. In *Proc. of EMNLP*.
- Federico, Marcello, 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 *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132.
- Gehring, Jonas, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 123–135.
- Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.
- Graves, Alex and Jürgen Schmidhuber. 2005. Frameworkwise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.
- Grbovic, Mihajlo, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, Ricardo Baeza-Yates, Andrew Feng, Erik Ordentlich, Lee Yang, and Gavin Owens. 2016. Scalable semantic matching of queries to ads in sponsored search advertising. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 375–384. ACM.
- Grbovic, Mihajlo, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. [E-commerce in your inbox: Product recommendations at scale](#). In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15*, pages 1809–1818, New York, NY, USA. ACM.
- Green, Spence, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. 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.
- Herbig, Nico, Santanu Pal, Mihaela Vela, Antonio Krüger, and Josef van Genabith. 2019. Multimodal indicators for estimating perceived cognitive load in post-editing of machine translation. *Machine Translation*, pages 1–25.
- Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hokamp, Christopher M. 2018. *Deep interactive text prediction and quality estimation in translation interfaces*. Ph.D. thesis, Dublin City University.
- Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.
- Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang,

- Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.
- Kenny, Dorothy. 2011. Electronic tools and resources for translators. In *The Oxford handbook of translation studies*.
- Kim, Junhong and Pilsung Kang. 2018. Recurrent neural network-based user authentication for freely typed keystroke data. *arXiv preprint arXiv:1806.06190*.
- Koehn, Philipp and Ulrich Germann. 2014. The impact of machine translation quality on human post-editing. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 38–46.
- Koponen, Maarit, Wilker Aziz, Luciana Ramos, and Lucia Specia. 2012. Post-editing time as a measure of cognitive effort. *Proceedings of WPTP*, pages 11–20.
- Lacruz, Isabel and Gregory M. Shreve. 2014. Pauses and cognitive effort in post-editing. *Post-editing of machine translation: Processes and applications*, page 246.
- Lacruz, Isabel, Gregory M Shreve, and Erik Angelone. 2012. Average pause ratio as an indicator of cognitive effort in post-editing: A case study. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 21–30. AMTA.
- Lowe, Ryan, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. [The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems](#). *CoRR*, abs/1506.08909.
- Lu, Yichao, Phillip Keung, Shaonan Zhang, Jason Sun, and Vikas Bhardwaj. 2017. A practical approach to dialogue response generation in closed domains. *arXiv preprint arXiv:1703.09439*.
- Maaten, Laurens. 2009. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, pages 384–391.
- Maaten, Laurens van der and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Monrose, Fabian and Aviel D Rubin. 2000. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359.
- Moorkens, Joss and Sharon O’Brien. 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*.
- O’Brien, Sharon. 2006. [Pauses as indicators of cognitive effort in post-editing machine translation output](#). *Across Languages and Cultures*, 7(1):1–21.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proc. of International Conference on Learning Representations*.
- Sanchez-Torron, Marina and Philipp Koehn. 2016. Machine translation quality and post-editor productivity. *AMTA 2016, Vol.*, page 16.
- Sin-wai, Chan. 2014. The development of translation technology. *Routledge Encyclopedia of Translation Technology*, page 3.
- Snover, Matthew, 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.
- Specia, Lucia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.
- Specia, Lucia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadina, Matteo Negri, , and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Ma-*

- chine Translation Summit XVI*, pages 55–71, Nagoya, Japan.
- Specia, Lucia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.
- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Tatsumi, Midori, Takako Aikawa, Kentaro Yamamoto, and Hitoshi Isahara. 2012. How good is crowd post-editing? its potential and limitations. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 69–77.
- Toral, Antonio, Martijn Wieling, and Andy Way. 2018. [Post-editing effort of a novel with statistical and neural machine translation](#). *Frontiers in Digital Humanities*, 5:9.
- Vaswani, Ashish, 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 6000–6010.
- Yang, Diyi, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010.
- Yin, Pengcheng, Graham Neubig, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. 2019. [Learning to represent edits](#). In *International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA.