Interactive Post-Editing for Verbosity Controlled Translation

Prabhakar Gupta[1], **Anil Nelakanti**[1], **Grant M. Berry**[2], **and Abhishek Sharma**[1]

[1]Amazon Prime Video
[1]{prabhgup,annelaka,naabhiss}@amazon.com
[2]Department of Spanish, Villanova University
[2]grant.berry@villanova.edu

Abstract

We explore Interactive Post-Editing (IPE) models for human-in-loop translation to help correct translation errors and rephrase it with a desired style variation. We specifically study verbosity for style variations and build on top of multi-source transformers that can read source and hypothesis to improve the latter with user inputs. Token-level interaction inputs for error corrections and length interaction inputs for verbosity control are used by the model to generate a suitable translation. We report BERTScore to evaluate semantic quality with other relevant metrics for translations from English to German, French and Spanish languages. Our model achieves superior BERTScore over state-of-the-art machine translation models while maintaining the desired token-level and verbosity preference.

1 Introduction

Recent machine translation (MT) models (Sutskever et al., 2014; Bahdanau et al., 2015) have shown to excel with aspects of translation quality like adequacy and fluency but these models still suffer notable shortcomings like out-of-domain data, low-resource languages, rare words and longer sentences (Koehn and Knowles, 2017). Hence, MT systems are often supplemented by human translators for editing and correcting MT outputs to achieve the desired quality bar for various use-cases (Peris et al., 2017). Broadly, MT employing human input can be classified as one of two types: manual post-editing (MPE) and interactive post-editing (IPE) (Escribe and Mitkov, 2021). MPE relies on humans to make all necessary edits on top of MT output to deliver the final translation. Whereas, IPE uses a human-in-the-loop approach: the model offers human translators various cues like auto-complete suggestions, word look-ups, etc until the human arrives at a translation. Both approaches have their trade-offs and it is observed that while MPE is slower, it delivers a higher quality output relative to the faster IPE (Green et al., 2014). Our work is a human-in-the-loop model that aims to make efficient use of human effort in delivering improved translations.

Translation quality, largely encompassing adequacy and fluency, has been the primary focus of most MT studies. Some recent work has explored other aspects of MT models like translation diversity, word choice (rare vs frequent words), translation style, etc (Wang et al., 2021; Niu and Carpuat, 2020; Agrawal and Carpuat, 2019; Marchisio et al., 2019; Lakew et al., 2019; Niu et al., 2018; Yamagishi et al., 2016). This style-aware modelling further broadens the scope and usability of MT, even more so when users can control levers to achieve the desired style variation. Our work identifies one of the most important style features – verbosity or translation length. Controlling length of translation output has been studied before for NMT models (Lakew et al., 2019) but to the best of our knowledge our work is the first to propose a solution for controlling length of translation in an interactive human-in-the-loop setting. Length is extremely crucial in many layout constrained translations use-cases like subtitling where the same amount of information needs to be available on-screen at a given point in time independent of the subtitle language.

In this work, we propose a interactive post-editing system that leverages multi-source transformers to offer users:

- Interactive control for corrections,
- Support for verbosity variation and corresponding translation customization; and
- Reduced human effort by providing alternative word and phrase choices.

2 Related Work

We review some recent studies focusing on automatic and interactive post-editing models germane to our work. Automatic Post-Editing (APE) is the task of automatically correcting the output of an (MT) system. APE models can be used to adapt a general purpose MT systems to new domains, fix errors in MT outputs and, in general, reduce human post-editing effort (Chatterjee et al., 2015). Transformer-based models for APE systems (Sharma et al., 2021; Yang et al., 2020; Chatterjee et al., 2020) have eclipsed models relying on statistical MT in recent years (Simard et al., 2007; Béchara et al., 2012). APE models inherit all the drawbacks of NMT since there is no human involved. They are excellent at domain adaption but often fail to improve the quality of state-of-the-art NMT models (Sharma et al., 2021) and the role and relevance of APE is often debated (do Carmo et al., 2021).

Post-editing models models require processing of both source text and MT output in order to generate a revised translation. Typically, two separate encoders are used – one for each source text and corresponding candidate translation – in addition to a single decoder responsible for generating output. We adapt one such model the Multi-Source Transformer (MST) (Tebbifakhr et al., 2018) for the current research. There are alternate methods that use two sequence-to-sequence models instead and merge the resulting distributions (Junczys-Dowmunt and Grundkiewicz, 2016). Merging distributions post-hoc is inadvisable, as complex patterns cannot be learned as easily and care must be taken to ensure that the combined distribution remains representative. We extend the MST approach for our work and use it train interactive human-in-loop models with user control to improve translation quality and style.

In contrast to standard post-editing models, IPE models consume user inputs to revise candidate translations. User inputs could be tokens that should either be dropped or retained from candidate or source sentences. QuickEdit model (Grangier and Auli, 2018) is an example that uses strike-out interactions to gather user tokens that should be dropped from MT output. Similarly, TouchEditing model (Wang et al., 2020) supports substitution, deletion, reordering and insertion operations on tokens. Support for richer and more complex interactions makes the model more flexible and easier for users. Our work borrows token-level interactions on source text and candidate sentences (we call them hypothesis) from related literature that help correct errors and improve translation quality. We extend this further to control verbosity of translation.

Style-aware language generation has drawn considerable attention from researchers recently. They have been studied for paraphrasing and translating text with the desired style properties. Style properties like (active/passive) voice (Yamagishi et al., 2016), formality (Niu et al., 2018; Niu and Carpuat, 2020), complexity (Agrawal and Carpuat, 2019; Marchisio et al., 2019), and verbosity (Lakew et al., 2019) are some examples that were explored in various applications. We evaluate one of the most important style features – verbosity or translation length in the context of IPE in this work.

3 Problem and Approach

Our primary goal is to train a sequence-to-sequence model that can improve candidate translations of the source with user cues. The model $\mathcal{M}(s, h, \mathcal{I})$ takes as its input a pair of sentences (source text s and a hypothesized translation h) as well as a set $\mathcal{I}$ of user interaction cues. The model tries to improve the translation of the source while leveraging user cues and accommodating those that are feasible. Post-edits can be successively applied by translators until satisfactory translation as $h_{k+1} = \mathcal{M}(s, h_k, \mathcal{I}_k)$.

Users generate interactions on the source and hypothesis sentences that the model uses to improve the translation. We support two categories of interactions; (1) token-level interactions (2) length interactions. For token-level interactions, the user has access to four unique operations $\mathcal{I} = \{keep, delete, insert, replace\}$. At each iteration, interactions can reflect any subset of this set $\mathcal{I}_k \subset \mathcal{P}(\mathcal{I})$, including an empty set $\emptyset$ which reduces to APE. The *keep* interaction is used to mark tokens in hypothesis that the user wishes to retain in the revised translation and the *delete* interaction captures tokens from the hypothesis the user prefers to drop from the revised translation (similar to the strike-out operation in QuickEdit). The *replace* interaction allows user to mark tokens in the hypothesis that are in the right position but need to be changed in the revised translation. For the *insert* interaction, we provide the user with a translation language model (TLM) (similar to (Con-

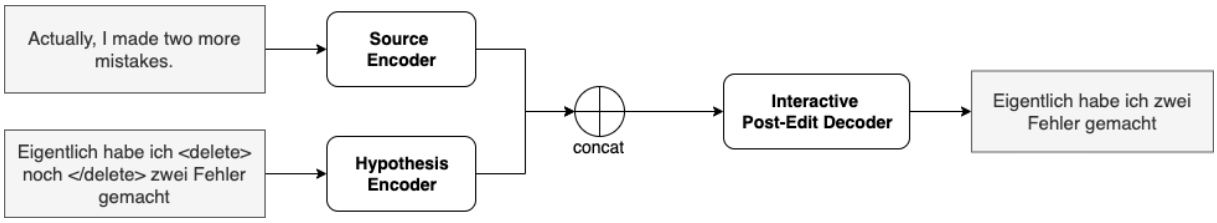

Figure 1: The model architecture. The source and the hypothesis are encoded using two separate transformer encoders. The resulting encodings are concatenated and given as the input to the decoder.

neau and Lample, 2019)) which allows the user to add a token in the hypothesis at a position of their choice. These interactions are passed by enclosing the relevant tokens from the sentence in appropriate control tags. For example, *keep* is indicated by using <keep> and </keep> tags around relevant tokens in the hypothesis. [1]

The length interaction is used to affect the verbosity of the output. User can request three possible variations on this interaction; increment (↑), decrement (↓) or isometric translation (↔). Isometric translation refers to generating translations similar in length to the source. We follow the same definition of isometric translation as described in (Lakew et al., 2021). Any translation with target character length in $\pm 10\%$ of source character length is considered isometric. If the user wants to make the revised translation longer/shorter compared to the hypothesis, they indicate this with the increment or decrement length interactions. If the user wants to make the revised translation to be of similar length as source, they pick isometric translation. The corresponding control tag is prepended to the hypothesis sentence before it is passed to the encoder. Table 1 shows examples of the various interactions described above.

Model architecture. We use a Multi-Source Transformer (MST) proposed by (Tebbifakhr et al., 2018) for our task. We specifically consider the implementation by (Wan et al., 2020) [2] based on Fairseq (Ott et al., 2019). The MST network consists of two transformer encoders and one transformer decoder (Vaswani et al., 2017). We use the same hyperparameters as in the Transformer (base) model with 6 layers in both the encoder and the decoder, 512 as the embedding dimension, 2048 as dimension of the feed forward layer and 8 heads for multi-head attention. The two encoders are used for encoding the source and the hypothesis sentences separately, while the decoder is used for generating an improved translation. Outputs from the encoders are concatenated and then used to generate keys and values for the encoder-decoder attention sub-layers in the decoder. (See Figure 1 for an illustration of the model.) We maximize the conditional log-likelihood $\mathcal{L}_\theta(D)$ of the training data D over the network parameters θ,

$$\theta^* = \underset{\theta}{\text{argmax}} \mathcal{L}_\theta(D) \equiv \sum_{(s,t,h,\mathcal{I})\in D} \log \mathrm{P}_\theta(t|s,h,\mathcal{I}),$$

where $\mathrm{P}_\theta(t|s,h,\mathcal{I})$ is probability of target sentence given the inputs that is auto-regressively estimated as $\prod_{k=1}^{|t|} \mathrm{P}_\theta(y_k|y_{k-1},s,h,\mathcal{I})$. The model $\mathcal{M}$ performs auto-regressive decoding with beam search using the distribution of the next token $\mathrm{P}_{\theta^*}(y_k|y_{<k},s,h,\mathcal{I})$ conditioned on the previously generated tokens $y_{<k}$, the model inputs, and optimal parameters θ^*. We use a beam width of 5 in our experiments and do not use any hard constraints during decoding.

4 Interaction Data Simulation

To the best of our knowledge, there are no publicly available datasets capturing the various interactions we aim to learn through $\mathcal{M}$. There are public datasets for automated post-editing (like (Chatterjee et al., 2019)) but they only have source and hypothesis pairs with post-edit required on the latter to improve the translation. They do not cover the spectrum of interactions we are considering in our work. Ideally, we would need an interface to collect the necessary data spanning our interaction set where professional translators can mark the edits necessary to arrive at a satisfactory translation from given inputs. This method of data collection is expensive and does not scale to larger sets of supported interactions. Alternatively, we can synthesize the input by simulating user interactions from

[1] The description of user interface used for these interactions is briefly described in Section 5.2 and other details beyond the scope of this work.

[2] https://github.com/zerocstaker/constrained_ape

Table 1: Examples of inputs (source s, hypothesis h, interaction $\mathcal{I}$) and corresponding model output (translation $\hat{t}$) along with human given reference translation (t). (A) Interaction was to keep two words: *"Deshalb"* and *"führend"* from the h and they were correctly kept in the $\hat{t}$. (B) Interaction was to delete one word: *"noch"* which means *"more"* in the context. The literal translation of s would require *"noch"* to be there but because of Delete interaction it was removed from $\hat{t}$. (C) Longer Interaction translated the 22 characters long h to a 43 characters long $\hat{t}$ while maintaining the meaning of translation. (D) Shorter Interaction translated 45 characters long h to a 21 characters $\hat{t}$. (E) Length of s is 48 characters making the 39 character long h non-isometric translation but generated $\hat{t}$ is 45 character long making it an isometric translation.

(A) Token Interaction - Keep
s: So I think we have to be in the lead.
h: <*keep*> Deshalb </*keep*> denke, wir <*keep*> führend </*keep*> sein.
$\hat{t}$: Deshalb denke ich, dass wir führend sein müssen.
t: Deshalb denke ich, müssen wir führend sein.
(B) Token Interaction - Delete
s: Actually, I made two more mistakes.
h: Eigentlich habe ich <*delete*> noch </*delete*> zwei Fehler gemacht.
$\hat{t}$: Eigentlich habe ich zwei Fehler gemacht.
t: Eigentlich machte ich zwei Fehler.
(C) Length Interaction - Longer
s: He kept pointing here.
h: <*length*↑> Er zeigte hier weiter.
$\hat{t}$: Er hat diesbezügliche Fortschritte gemacht.
t: Er zeigte immer hier hin.
(D) Length Interaction - Shorter
s: It's a complete denial of mistakes.
h: <*length*↓> Es ist eine völlige Verweigerung von Fehlern.
$\hat{t}$: Es ist völlig falsch.
t: Es ist eine total Verleugnung von Fehlern.
(E) Length Interaction - Isometric
s: Is it something about the details or the colors?
h: <*length*↔> Geht es um die Details oder die Farben?
$\hat{t}$: Geht es um die Einzelheiten oder die Farben?
t: Geht es dabei um die Details oder die Farben?

bilingual parallel text data that is readily available. We take this route similar to various post-editing studies (Grangier and Auli, 2018; Tebbifakhr et al., 2018) to identify and mark tokens with interactions that the model can leverage to improve the hypothesis.

Interaction simulation. We begin with a set of high quality data bitext data samples $(s, t) \in S$ for the language pair of interest. A high-quality pre-trained machine translation system T_S is used to generate hypotheses $h = T_S(s)$. User interactions are then simulated from the hypothesis and reference pairs as $\mathcal{I} = U(h, t)$. This gives us the input triplets $(s, h, \mathcal{I})$ to train our model. Following are the details of the simulation function U that imitates translator interactions from (h, t) sentence pairs.

- **Delete tokens.** The *delete* interaction allows user to specify token substrings in the hypothesis that should be dropped from the final translation. It helps the model learn to correct over-translated phrases and rephrase translation accordingly. We simulate *delete* interaction by sampling from substrings in h do not appear in the corresponding t.
- **Keep tokens.** The *keep* interaction has the opposite effect of *delete* and is used to retain tokens from the hypothesis that are good translations of the corresponding source. Contrary to *delete*, we sample from substrings from h that match t to identify relevant tokens.
- **Replace or Insert tokens.** The *replace* interaction allows user suggest replacements to

some tokens in the hypothesis. These suggestions could be generated from other models like a masked language model or manually entered by users. For training, we sample the substrings from h that do not appear in the t but there is an acceptable replacement in the t. We change the hypothesis by introducing the replacement tokens in the hypothesis within <keep> tags. Insert token operation operates similarly except that while replace token operation applies to a token in the hypothesis, insert token can be used in between tokens.

- **Length Interaction.** This interaction is useful in changing the verbosity of the translation candidates relative to h or s. The user can specify whether the output should be *longer* or *shorter* than h or in same range compared to s. The *length* interaction can be of very high interest in domains where the length of the final translation is crucial. For example, movie subtitle translations impose display limitations on screen that constrain the length of the text (Gupta et al., 2019). Using $|\cdot|$ to refer to character length of a sentence and a hyperparameter $\delta_l = 0.1$, we mark the interaction as *longer* when the ratio $(|t| - |h|)/|t| > \delta_l$, *shorter* if $(|t| - |h|)/|t| > -\delta_l$ and *isometric* if $(|t| - |s|)/|s| < \delta_l$ and $(|h| - |s|)/|s| > \delta_l$. The samples that fall in neither bucket are marked with *no-preference*.

We ignore all candidate substrings with more than eight tokens for token-interactions and skip any substrings that only have punctuation tokens. A token can belong to no more than one interaction and for a given input triplet multiple operations can be sampled across token substrings. We do not limit the number of token-interactions that are present in a sample but we control the number of interactions by occurance probabilities of each interaction. To make the model more robust, for each sample there was a 5% chance of getting a noisy token-level interaction of random length. For length interaction, again there was a 5% chance of prepending a random interaction from *longer*, *shorter*, *isometric* or *no-preference*. We compute the interactions for each sample on-the-fly and because of the random chances we have introduced in interaction simulation, we are able to train the model with multiple versions of same sample.

At inference time, user can provide any or all of the length interaction and token-level tags to the model. Table 1 shows examples for each of the described interactions.

5 Experiments and Results

Train and Test data. We use two public parallel datasets for training the model – European Parliament Proceedings Parallel Corpus (EuroParl) (Koehn, 2005) and MuST-C dataset (Cattoni et al., 2021). Both datasets represent high quality parallel corpora. MuST-C is a multilingual speech translation corpus with hundred hours of audio recordings from English TED Talks, which are automatically aligned at the sentence level with their manual transcriptions and translations. We do a 99-1 split of dataset into train, validation sets. We use two public test datasets – FLORES (Goyal et al., 2021) and MuST-C test set provided with dataset. FLORES is a high-quality many-to-many multilingual translation benchmark dataset for 101 languages. Table 2 provides the distribution statistics of each interaction of our train-validation-test sets by target language.

Table 2: Train and Test dataset sizes

Dataset Name	Split	En-De	En-Es	En-Fr
EuroParl	Train	1.7M	1.81M	1.77M
MuST-C		217K	250K	257K
FLORES	Test	997	997	997
MuST-C		2641	2502	2632

We use pretrained machine translation models, OPUS-MT (Tiedemann and Thottingal, 2020) to generate a hypothesis from source. For En-De, we use an additional model for generating hypothesis – FAIR's submission for WMT19 news translation task (Ng et al., 2019). We use BERT (Devlin et al., 2019) tokenizer provided in HuggingFace Tokenizer[3] to tokenize our inputs instead of generating a new dictionary. We clean the data by removing samples with more than 250 tokens.

We chose the Adam optimiser (Kingma and Ba, 2015) with $(\beta_1, \beta_2) = (0.9, 0.98)$ and an initial learning rate of $5e$-4 with inverse square-root decay scheduled after 4000 warm-up steps. We reduce the number of trainable parameters by using joint vocabulary for source, hypothesis and target, and by sharing the input and output embedding for the decoder. For regularisation, we use dropout (Srivastava et al., 2014) with a value 0.3 and weight decay

[3] https://huggingface.co/bert-base-multilingual-cased

Table 3: BERTScore of do nothing baseline (OPUS-MT) compared against our model with no interactions (APE condition) and all interactions (IPE model). More detailed scores in Table 4

Dataset	Model	En-De	En-Es	En-Fr
FLORES	Do Nothing Baseline	0.887	0.875	0.914
	No Interactions	0.881	0.87	0.907
	All Interactions	**0.922**	**0.923**	**0.94**
MuST-C	Do Nothing Baseline	0.877	0.89	0.903
	No Interactions	0.878	0.891	0.902
	All Interactions	**0.926**	**0.934**	**0.941**

of $1e$-4. To train the model, we use label-smoothed cross-entropy criterion with the smoothing parameter set to 0.1. For each language-pair we train a different model. We train each model for 24 epochs with one dataset per epoch and choose the checkpoint that gives the lowest loss on the validation set.

5.1 Results and Observations

We report results on model generated translations on the test $\hat{t} = \mathcal{M}(s, h, \mathcal{I})$ and focus our study on two specific aspects; constraint satisfaction (CS) and translation quality (TQ). For each interaction, we define the constraint satisfaction criterion in respective subsections. Translation quality is evaluated using the BERTscore (Zhang et al., 2020). BERTScore correlates better to the human judgement than its predecessors like BLUE (Papineni et al., 2002).

As reported in Table 3, the baseline BERTScore(h, t) wherein h are translations from OPUS-MT models is at 0.892 when averaged across three language pairs for FLORES test set and at 0.89 for the MuST-C test set. Using the full interaction set $\mathcal{M}(s, h, \mathcal{I})$ gives 0.92 and 0.928 BERTScore on FLORES and MuST-C respectively averaged over three language pairs. For **Do Nothing Baseline**, $\mathcal{M}(s, h, \emptyset)$ with no interaction during inference gives 0.886 and 0.877 BERTScore on both sets respectively averaged over three language pairs. This is similar to automatic post-editing (APE) where the model receives no human intervention. Previous works have shown in past that outperforming high-quality NMT models for APE with general data is extremely difficult (Sharma et al., 2021) and we observe a similar trend with our results.

Token Level Interactions. User inputs act like soft constraints and are not hard enforced hence the translation output does not always conform with the interaction request. We do a study to evaluate if the token-level controls also reflect similarly in the model outputs $\hat{t}$ as intended by the specified user interaction. We only report the numbers for *keep* and *delete* interactions since *insert* and *replace* interactions are essentially just *keep* interactions for the model as explained in Section 4. For the *keep* interactions, we check what percentage of tokens marked in h appear in the output translation $\hat{t}$. Similarly, for the *delete* interaction, we calculate the percentage of tokens that were marked in h for deletion and do not appear in $\hat{t}$. Table 4 reports corresponding results showing a high level of agreement in generated $\hat{t}$ (90% for *keep* and 60% for *delete*) with user requests. We only report the numbers for samples where at least one such interaction was possible, we skip other sentences where that interaction was not feasible. We saw a further improvement in performance when we used these interactions together as compared to just one interaction at a time (93% for *keep* and 65% for *delete*).

Length Interactions. For length interactions, it is more important to understand if the user request for change in verbosity is met with. Correspondingly, we compute change in verbosity between h and $\hat{t}$ to evaluate the affect of style controls. We take a ratio of character lengths $|\hat{t}|/|h|$ and report averages separately for *No preference*, *Longer* and *Shorter* control options. The results, summarized in Table 4, clearly show the intended trend. The model's ability to manipulate verbosity without negatively impacting translation quality (as measured by BERTScore) is evident.

Isometric Translation. Contrary to other length interactions, where the verbosity is relative to the length of h, for Isometric Translation verbosity is determined with respect to the length of s. With this in mind, we take a ratio of token lengths $|\hat{t}|/|s|$

Table 4: Constraint Satisfaction (CS) and Translation Quality (TQ) (1) Token Level Interactions: CS is tokens % marked with the interaction in hypothesis h and were satisfied in output translation $\hat{t}$. (2) Length Interactions: CS is the average ratio of character length of $\hat{t}$ and the that of h. (3) Isometric Translation: CS is the average ratio of character length of $\hat{t}$ and the that of s. No Preference is the ratio of t and s. *Custom NMT* is the NMT model we trained to specifically generate isometric translations. For TQ, we use BERTScore (Zhang et al., 2020).

		En-De		En-Es		En-Fr	
		CS	**TQ**	**CS**	**TQ**	**CS**	**TQ**
Token Level Interactions							
FLORES	Keep	0.913	0.879	0.928	0.87	0.921	0.906
	Delete	0.581	0.882	0.584	0.871	0.631	0.908
MuST-C	Keep	0.961	0.878	0.976	0.892	0.967	0.902
	Delete	0.586	0.879	0.605	0.893	0.606	0.904
Length Interactions							
FLORES	No Preference	0.92	0.881	0.923	0.87	0.926	0.907
	Longer	0.941	0.88	0.949	0.869	0.941	0.906
	Shorter	0.82	0.864	0.831	0.857	0.872	0.897
MuST-C	No Preference	0.865	0.878	0.868	0.891	0.869	0.902
	Longer	0.898	0.878	0.92	0.886	0.9	0.902
	Shorter	0.775	0.862	0.79	0.881	0.823	0.895
Isometric Translation							
FLORES	No Preference	1.174	0.887	1.199	0.875	1.193	0.914
	Custom NMT	0.932	0.823	1.051	0.851	1.035	0.875
	Ours	1.069	0.877	1.084	0.868	1.137	0.906
MuST-C	No Preference	1.121	0.877	0.998	0.89	1.147	0.903
	(Lakew et al., 2019)	1.02	-	-	-	-	-
	Custom NMT	0.961	0.833	0.999	0.883	1.053	0.895
	Ours	1.048	0.872	1.023	0.887	1.107	0.903

and report averages for each language-pair in Table 4. Model's ability to generate isometric translations while maintaining the translation quality (as measured by BERTScore) is evident from these empirical results. Ideally CS ratio, should be in range $[0.91, 1.1]$; anything between this range would be considered isometric.

We compare our model with the small data condition for En-De described in (Lakew et al., 2019). The *match* scenario described by the authors is the very similar to the isometric translations. They used a similar dataset as us to train an NMT model to exclusively generate isometric translations. Of the languages we are considering in this work, they only report their performance for En-De model on MuST-C dataset. A direct comparison of CS ratio is unfair since authors tried to generate translations as closely matching in length with the source while we generate translations that are isometric. For TQ, they report BLEU scores (Papineni et al., 2002). They report 27.60 BLEU points while we achieve 34.15 BLEU points for En-De pair.

We trained another version of our model with no hypothesis to demonstrate the importance of hypothesis in generating a good translation with interactions. Instead of passing the hypothesis to the Hypothesis Encoder, we only pass the length control token and let the concatenation occur similar as in the original model. This allows the model to generate a token-level embedding for the control token which is concatenated to the source embedding. The results are reported as *Custom NMT* in Table 4. Some recent work have attempted to do this using positional embedding to pass length control information. Authors of (Takase and Okazaki, 2019) did this for text summarization and (Lakew et al., 2019) attempted it for generating length controlled translations calling it *Length Encoding* method.

Comparing *Ours* model with other approaches, we can see our model is able to generate better quality translations than a dedicated isometric translation model while providing access to multiple other interactions as well.

5.2 User Study

To study the feasibility of the our approach, we conducted user trails with five translators. All translators were proficient in two languages; English and one additional language. We conducted the experiment with models trained similarly but with English as target language to make it easier for us to analyse the results. Two translators worked on Italian [4] as source language, two on Spanish and one on French [5].

Each translator was provided the same hand-picked sentences in the same order from FLORES dataset with hypothesis generated from OPUS-MT (Tiedemann and Thottingal, 2020). Each translator was asked to work on as many translations as possible in 30 minutes. Users were shown a source sentence in English and hypothesis in the language of their proficiency. They could then accept a translation or provide the interactions to improve the quality of the translation. Providing interactions generates a revision using the Interactive Post-Edit (IPE) model we have trained; the user can again either accept or revise the revision. After four revisions, users had an option to enter the translation manually.

We found around 59% of the hypothesis were accepted without any revisions verifying that the OPUS-MT translation model used is already of high quality. Users employed the IPE model in 13% of cases to make quick edits. For the remaining samples, users tried multiple revisions and after an average of 3.57 revisions, users preferred writing the translations manually. We saw a minor improvement in quality of translations with the IPE model as well; showing the model's capability to generate translations more efficiently without compromising the quality of the translations.

6 Conclusion

We propose and evaluate a model for human-in-loop interactive MT. The model offers the user controls that can be leveraged to correct mistranslations and rephrase them to achieve desired style variations. We specifically evaluate how the five interactions of *keep*, *delete*, *replace*, *insert*, and *length* perform in terms of translation quality as measured by BERTScore and interaction constraint satisfaction in final translation. User input remains the gold standard for ensuring translation quality, and providing user interactions enables human input when necessary to boost performance. Further, the empirical verification of the use of interactive control beyond translation corrections (as is common with existing post-editing models) to achieve desired style variations can serve as a major boost to customizability of MT systems.

As part of our future work, we wish to expand this study to more language pairs with multilingual models for reduced operational load and evaluate more style variations. We wish to go beyond explicit style tokens and use a continuous space for representing style edits on which we condition the decoder to generate a translation with corresponding variation. Such a representation is likely to be better suited to capture all style variations from data unsupervised without explicit labeling and tagging as we did with verbosity and readability.

[4] Even though we do not report results for Italian model in this work, the model used for user-study was trained as described for other languages

[5] There was one additional translator for French but we omitted the results since they did not understand the experiment and quit after couple of translations.

References

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hanna Béchara, Raphaël Rubino, Yifan He, Yanjun Ma, and Josef van Genabith. 2012. An evaluation of statistical post-editing systems applied to RBMT and SMT systems. In *Proceedings of COLING 2012*, pages 215–230, Mumbai, India. The COLING 2012 Organizing Committee.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Comput. Speech Lang.*, 66:101155.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 13–30, Florence, Italy. Association for Computational Linguistics.

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

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the planet of the APEs: a comparative study of state-of-the-art methods for MT automatic post-editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Félix do Carmo, Dimitar Shterionov, Joss Moorkens, Joachim Wagner, Murhaf Hossari, Eric Paquin, Dag Schmidtke, Declan Groves, and Andy Way. 2021. A review of the state-of-the-art in automatic post-editing. *Mach. Transl.*, 35(2):101–143.

Marie Escribe and Ruslan Mitkov. 2021. Interactive models for post-editing. In *Proceedings of the Translation and Interpreting Technology Online Conference*, pages 167–173, Held Online. INCOMA Ltd.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

David Grangier and Michael Auli. 2018. QuickEdit: Editing text & translations by crossing words out. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 272–282, New Orleans, Louisiana. Association for Computational Linguistics.

Spence Green, 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, Doha, Qatar. Association for Computational Linguistics.

Prabhakar Gupta, Mayank Sharma, Kartik Pitale, and Keshav Kumar. 2019. Problems with automating translation of movie/tv show subtitles. *CoRR*, abs/1909.05362.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers, MTSummit 2005, Phuket, Thailand, September 13-15, 2005*, pages 79–86.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Surafel Melaku Lakew, Marcello Federico, Yue Wang, Cuong Hoang, Yogesh Virkar, Roberto Barra-Chicote, and Robert Enyedi. 2021. Machine translation verbosity control for automatic dubbing. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7538–7542.

Surafel Melaku Lakew, Mattia Antonino Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. *CoRR*, abs/1910.10408.

Kelly Marchisio, Jialiang Guo, Cheng-I Lai, and Philipp Koehn. 2019. Controlling the reading level of machine translation output. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 193–203, Dublin, Ireland. European Association for Machine Translation.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319.

Xing Niu and Marine Carpuat. 2020. Controlling neural machine translation formality with synthetic supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8568–8575.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Comput. Speech Lang.*, 45:201–220.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online. Association for Computational Linguistics.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3999–4004.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer with combined losses for automatic post editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 846–852, Belgium, Brussels. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT - building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisboa, Portugal, November 3-5, 2020*, pages 479–480.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David Wan, Chris Kedzie, Faisal Ladhak, Marine Carpuat, and Kathleen McKeown. 2020. Incorporating terminology constraints in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1193–1204, Online. Association for Computational Linguistics.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11, Suzhou, China. Association for Computational Linguistics.

Yue Wang, Cuong Hoang, and Marcello Federico. 2021. Towards modeling the style of translators in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1193–1199, Online. Association for Computational Linguistics.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. HW-TSC's participation at WMT 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.