QuickEdit: Editing Text & Translations via Simple Delete Actions

David Grangier **Michael Auli**
Facebook AI Research
Menlo Park, CA

Abstract

We propose a framework for computer-assisted text editing. It applies to translation post-editing and to paraphrasing and relies on very simple interactions: a human editor modifies a sentence by marking tokens they would like the system to change. Our model then generates a new sentence which reformulates the initial sentence by avoiding the words from the marked tokens. Our approach builds upon neural sequence-to-sequence modeling and introduces a neural network which takes as input a sentence along with deleted token markers. Our model is trained on translation bitext by simulating post-edits. Our results on post-editing for machine translation and paraphrasing evaluate the performance of our approach. We show +11.4 BLEU with limited post-editing effort on the WMT-14 English-German translation task (25.2 $\rightarrow$ 36.6), which represents +5.9 BLEU over the post-editing baseline (30.7 $\rightarrow$ 36.6).

1 Introduction

Computers can help humans edit text more efficiently. In particular, statistical models are used for that purpose, for instance to help correct spelling mistakes (Brill and Moore, 2000) or suggest likely completions of a sentence (Bickel et al., 2005). In this work, we rely on statistical learning to enable a computer to rephrase a sentence by only pointing at words that should be avoided. Specifically, we consider the task of reformulating either a sentence, i.e. *paraphrasing* (Quirk et al., 2004), or a translation, i.e. *translation post-editing* (Koehn, 2009b). Paraphrasing reformulates a sentence with different words preserving its meaning, while translation post-editing takes a candidate translation along with the corresponding source sentence and improves it.

Our proposal relies on very simple interactions: a human editor modifies a sentence by selecting tokens they would like the system to replace and no other feedback. Our system then generates a new sentence which reformulates the initial sentence by avoiding to use the words from the selected tokens. Our approach builds upon neural sequence-to-sequence and introduces a neural network which takes as input a sentence along with deleted token markers. We introduce a novel attention-based architecture suited to this goal and propose a training procedure based on simulated post-edits on translation bitext (§3). This approach allows to get substantial modifications of the initial sentence – including deletion, reordering and insertion of multiple words – with limited user effort.

Our experimental results (§4) show that our model outperforms the considered post-editing baseline by up to 5 BLEU points on WMT'14 en-de and WMT'14 de-en. The advantage of our method is also highlighted in monolingual settings.

In the rest of the paper, we first describe related work (§2), we then introduce our method (§3) and evaluates its performance on post-editing and monolingual editing (§4). Finally, we draw some conclusions and delineate potential future research directions (§5).

2 Related Work

Our work builds upon previous research on neural machine translation, machine translation post-editing, and computer-assisted editing.

2.1 Neural Machine Translation

Statistical machine translation systems builds models to automatically translate text relying on large corpora of bitext, i.e. corresponding pairs of sentences in the source and target language (Koehn, 2009a). Recently, machine translation systems based on neural networks have emerged as an effective approach to this problem (Sutskever et al., 2014). Neural networks are a departure from count-based translation systems, phrase-based systems, which used to dominate the field (Koehn, 2009a).

Research in Neural Machine Translation (NMT) focuses notably on identifying appropriate neural architecture. Cho et al. (2014) and Suskever et al. (2014) proposed an encoder/decoder model. These models consist in a Recurrent Neural Network (RNN) mapping the source sentence sentence into a latent vector (encoder). This vector conditions an RNN language models (decoder) which generates the target sentence (Mikolov et al., 2010; Graves, 2013). Bahdanau et al. (2014) adds attention to these models, which leverages the fact that the explanation for a given target word in generally localized around a few source words. Recently, new architectures have proposed to replace recurrent modules with convolutions (Gehring et al., 2017) or self-attention (Vaswani et al., 2017) to further increase accuracy. These architecture also perform attention at more than one decoder layers, allowing for more complex attention patterns. In this work, we build upon the architecture from Gehring et al. (2017) since this model offers a good trade-off between high accuracy and fast decoding.

2.2 Translation Post-Editing

Post-editing leverages a machine translation system and enable human translators to edit its output with different levels of computer assistance. This enables improving machine translation outputs with lesser effort than purely manual translation.

Green et al. (2014) implement such a system relying on a phrase-based translation system. The system presents an initial translation to the user who can accept a prefix and select among the most likely postfix iteratively. Similar ideas relying on decoding with prefix constrains are common in post-translation (Langlais et al., 2000; Koehn, 2009b; Barrachina et al., 2009). Similar, approaches based on left-to-right decoding have been extended to neural machine translation recently (Peris et al., 2017).

Closer to our work, Marie and Max (2015) proposes light-weight interactions based on accepting/rejecting spans from the proposed machine translation. The user labels each span that need to appear in the final translation, doing so they also reject all spans which are not positively labeled. The system then removes the rejected spans from the phrase table. It also edits phrase pairs out of the phrase table such that the positively marked target spans are the only ones allowed to explain the corresponding source phrases. Compared to this work, we rely on similar interactions but we do not require the user to label every token as accepted or rejected. In our case, the user only need to mark a few rejections. More importantly, our approach is built on top of a more accurate neural MT system which is not amenable to phrase table editing. Finally, our method is also applicable to a monolingual setting to edit of regular text.

2.3 Computer-Assisted Text Editing

Computer assisted text editing has been introduced with interactive computer terminals (Irons and Djorup, 1972). Its first achievement was to simplify the insertion, deletion, and copy of text compared to typewriters. Computers then enabled the emergence of computerized language assistance tools such as spelling correctors (Brill and Moore, 2000) or next word suggestions (Bickel et al., 2005).

More recently, research has focused on generating paraphrases (Bannard and Callison-Burch, 2005; Mallinson et al., 2017), compressing sentences (Rush et al., 2015) or simplifying sentences (Nisioi et al., 2017). This type of work expands the possibilities for *interactive* text generation tools, like our work.

Related to our work, Filippova et al. (2015) considers the task of predicting which tokens can be removed from a sentence without modifying its meaning relying on a recurrent neural network. This is different from our work since our model does not predict which token to remove, as the user gives them. Our generation is also more involved as our model rephrase the sentences, which includes introducing new words. Guu et al. (2017) considers generating text with latent edition. Their goal is not to enable users to control which words need to be rejected or accepted in an initial sentence but to enable sampling valid English sentences with high lexical overlap around a starting sentence.

3 QuickEdit

QuickEdit is our sequence-to-sequence model for post-editing via delete actions. This model takes as input a source sentence and an initial guess target sentence annotated with rejected tokens. It can then decode a new target sentence taking the rejection labels into account.

3.1 Model Architecture

This model builds upon the architecture from Gehring et al. (2017). Compared to this initial model, our model adds a second encoder to encode the annotated guess sentence. It also duplicate every attention layer to allow the decoder to attend both to the source and the guess sentences. Dual attention has been introduced recently in the context of automatic post-editing (Novak et al., 2016; Libovickỳ and Helcl, 2017).

The encoder of the initial guess takes as input a target sentence t annotated with binary rejection labels r, i.e.

$$g = \{g_i\}_{i=1}^{l_g} \text{ where } \forall i, g_i = (t_i, r_i)$$

in which l_g denotes the length of the guess, t_i is an index in the target vocabulary and r_i is a binary variable with 1 indicating a rejection by the user and 0 indicating no user preference. The first layer of the encoder maps this sequence to two embedding sequences, i.e. a sequence of target word embeddings and a sequence of positional embeddings. Compared to (Gehring et al., 2017), we extend the positional embedding to contain two types of vectors, positional vectors associated with positions i where $r_i = 0$ and positional vectors associated with positions i where $r_i = 1$. Like all parameters in the system, both sets of embeddings are learned to maximize the likelihood of the reference sentences conditioned on the source, annotated guess pairs.

The attention over two sentences is simple. Both source and guess encoders produce a sequence of key and value pairs. We denote the output of the source encoder as $\{(k_i^s, v_i^s)\}_{i=1}^{l_s}$ and the output of the guess encoder as $\{(k_i^g, v_i^g)\}_{i=1}^{l_g}$. At each decoder layer k and time step j, the decoder produces a latent state vector h_j^k, this vector attends to the output of the source encoder,

$$a_i^s = \exp\left(h_j^k \cdot k_i^s\right) / \sum_l \exp\left(h_j^k \cdot k_l^s\right)$$

and the guess encoder,

$$a_i^g = \exp\left(h_j^k \cdot k_i^g\right) / \sum_l \exp\left(h_j^k \cdot k_l^g\right).$$

This attention weights are used to summarize the values of the source $\sum_i a_i^s v_i^s$ and the guess $\sum_i a_i^s v_i^g$ respectively. The attention module then averages these two vectors $\frac{1}{2}\sum_i a_i^s v_i^s + \frac{1}{2}\sum_i a_i^g v_i^g$ and uses this average instead of the source attention output in the next layer (Gehring et al., 2017).

3.2 Training & Inference

Our model is trained on translation bitext by simulating post-edits. Given a bitext corpus, we first train an initial translation system and we then rely on this system to translate the training corpus. This strategy results in three sentences for each example: the source, the guess (i.e. the sentence decoded from the initial system) and the reference sentence. We then mark the guess tokens which do not appear in the corresponding reference sentence as rejected.

The dual attention model presented in the above section is then trained to maximize the likelihood of each reference sentence given the source and the annotated guess. Training relies on stochastic gradient descent (Bottou, 1991), using Nesterov's accelerated gradient with momentum (Nesterov, 1983; Sutskever et al., 2013). At inference time, we decode through standard left-to-right beam search (Sutskever et al., 2014). Our decoding strategy for QuickEdit also incorporates hard constraints forcing the decoder to avoid the tokens rejected from the guess.

3.3 Extension to Monolingual Editing

The extension of QuickEdit to monolingual setting is straightforward: we remove the source encoder and the corresponding attention path. This results in a single encoder model which takes only an annotated guess as input. This model can be trained from pairs of sentences consisting of a machine translation output along with the corresponding reference sentence. Although machine translation bitext are used to create this model training data, it operates solely on target language sentences without requiring a source sentence at test time.

4 Experiments & Results

We report results on IWSLT'14 German to English, IWSLT'14 English to German (Cettolo et al., 2014), WMT'14 German to English and WMT'14 English to German (Luong et al., 2015). As a baseline, we consider decoding with the initial translation system under the search constraints that hypotheses with rejected words cannot be selected in the beam.

For IWSLT'14 we train on 160K sentence pairs, and use a random subset of 7,250 sentences from the original training corpus as validation set. We test on the concatenation of *tst2010, tst2011, tst2012, tst2013, dev2010* and *dev2012* comprising 6750 sentence pairs. The vocabulary for this dataset is 24k for English and 36k for German. For WMT'14 we use the same setup as Luong et al. (2015) which comprises 4.5M sentence pairs for training and we test on newstest2014.[1] We took 45k sentences out of the training set for validation purpose. As vocabulary, we learn a joint source and target byte-pair encoding (BPE) with 44k types (Sennrich et al., 2016b,a). Note that even when using BPE, we solely rely on full word deletion labels, i.e. all the BPE tokens of a word carry the same binary deletion marker.

The model architecture settings are borrowed from (Gehring et al., 2017). For IWSLT'14 de-en and IWSLT'14 en-de, we rely on 4-layer encoders and 3-layer decoders, both with 256 hidden units. The word embedding for source and target as well as the output matrix have 256 dimensions. For WMT14-en-de and WMT14-de-en, both encoders and decoders have 16 layers, with 512 hidden units except for the last 2 layers which have 1,024 and 2,048 units respectively. All word embeddings have 768 dimensions. For all datasets, we decode using beam search with a beam of size 5.

4.1 Post-editing

Our study is based on simulated post-edits, i.e. simulated token deletion actions. We start from machine translation outputs from an initial system in which we label rejected words automatically. For initial translation, we rely on the convolutional translation system from (Gehring et al., 2017)[2] learned from the training portion of the dataset. For each system output, any word which does not belong to the reference translation is marked as rejected. We perform this operation for the train, validation and test portion of each dataset. The training and validation portion can be used for learning and developing our post-editing system. The test portion is used for evaluation. Table 1 reports our result on this task. Our quick-edit method strongly outperform the baseline post-editing system. On the larger WMT benchmark, the advantage is over 5 BLEU point for both direction. We conjecture that the improvement is lesser on the small set due to over-fitting, i.e. the base system is excellent on the training set which reduces the post-editing opportunities on the training data, therefore limiting the amount of supervised data for training our post-editing system. We show examples of post-editing from the test set of WMT-14 de-en in Table 2. These examples show the ability of the model to rephrase sentences avoiding the rejecting tokens while preserving the source meaning.

4.2 Monolingual Editing

Table 1 also reports monolingual results. In that case, the system is not given the source sentence, only a sentence in the target language along with rejection markers. Even if the model is not exposed to the source, it manage the generate sentences which are closer to the reference than the initial sentences, as shown by the BLEU improvement. This shows the ability of the model to paraphrase under deletion constraints. Table 3 shows example of the system in action from the English test set of WMT-14 de-en.

4.3 Partial Feedback

So far, our post-editing setting considers that all words needed to be removed from the initial translation are marked as such. We now consider the setting where the post-editor performs less work and only marks a subset of these words as rejected. This is analogous to a hypothetical online translation service which offers a feature enabling the user to progressively mark the parts of a translation which needs to improved. Figure 1 plots BLEU as a function of the number of word marked for rejection on the validation set from WMT'14 German to English. This curve is obtained by marking at most 1, 2, ..., 8 words for deletion per sentence, taking into account that the actual number of marked word in a sentence cannot be higher than the number of hypothesis words not present in the reference sentence. For this setting, we also trained models which were supervised with partial feedback. QE25, QE50, QE100 refer to Quick-Edit models trained with data where

[1] http://nlp.stanford.edu/projects/nmt

[2] https://github.com/facebookresearch/fairseq-py.

	IWSLT		WMT	
	de→en	en→de	de→en	en→de
initial translation	27.4	24.2	29.7	25.2
post-edit baseline	33.0	30.2	34.6	30.7
post-edit QuickEdit	34.6	30.8	41.3	36.6
monolingual QuickEdit	29.3	26.7	39.5	34.2

Table 1: Editing results (BLEU4) when all incorrect tokens are marked as rejected.

source	Schauspieler Orlando Bloom hat sich zur Trennung von seiner Frau, Topmodel Miranda Kerr, geäußert.
guess	Actor Orlando Bloom ~~has spoken of the~~ separation ~~of~~ his wife, ~~Topmodel~~ Miranda Kerr.
output	Actor Orlando Bloom **spoke about** separation **from** his wife, **Top Model** Miranda Kerr.
source	Die heutigen elektronischen Geräte geben im Allgemeinen wesentlich weniger Funkstrahlung ab als frühere Generationen.
guess	Today's electronic devices generally ~~give far less~~ radio ~~radiation~~ than previous generations.
output	Today's electronic devices generally **emit significantly fewer** radio **frequencies** than previous generations.
source	Boeing bestreitet die Zahlen von Airbus zu den Sitzmaßen und sagt, es stehe nicht im Ermessen der Hersteller zu entscheiden , wie Fluggesellschaften die Balance zwischen Flugtarifen und Einrichtung gestalten.
guess	Boeing is ~~denying~~ the figures ~~from~~ Airbus to the ~~seats~~ and says that it is not ~~left to the discretion of~~ the manufacturers to decide how airlines ~~are~~ to balance air fares and ~~set~~ up .
output	Boeing is **contesting** Airbus**'s seating** figures and says it is not **up to** manufacturers to **determine** how airlines balance fares and **equipment**.

Table 2: Post-editing examples from WMT'14 en-de. Strike-through text indicates the rejected words.

respectively 25, 50 or 100% of the hypothesis tokens not present in the reference were labeled as rejected.

Figure 1 shows a thin advantage for quick edit for 1-2 deleted words and larger improvement over the baseline for more substantial deletions. Unsurprisingly, the model trained with fewer deleted words (QE25, QE50) are more advantageous when tested with fewer deleted words, while QE100 gives the largest BLEU gain with 4 or more rejected words.

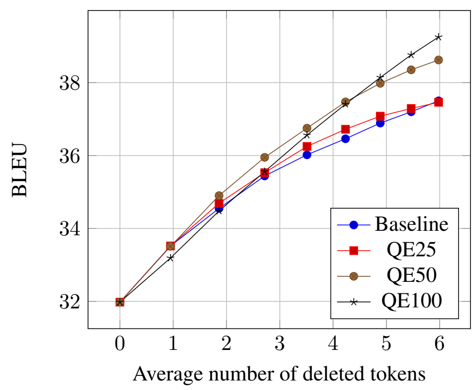

Figure 1: Post-editing results as a function of the average number of rejected tokens per sentence on WMT'14 de-en validation set (45k sentences). QE25, QE50, QE100 refer to Quick-Edit models trained with data where respectively 25, 50 or 100% of the hypothesis tokens not present in the reference were labeled as rejected.

5 Conclusions

This work proposes QuickEdit a sequence to sequence model that allow one to edit text by simply marking initial tokens as deleted. From a marked sentence, the model can generate an edited sentence both in the context of machine translation post-editing (a source sentence is also provided), or in a monolingual setting. In both cases, we assess the impact of the deleted action and shows that deleted words not present in a hidden reference sentence allow the model to generate text closer to this reference.

input	In an interview with Martin , Daley confirmed that the ~~government had actually considered~~ replacing Biden with Clinton.
output	In an interview with Martin , Daley confirmed that the **administration did indeed consider** replacing Biden with Clinton.
input	NSA revelations ~~reinforce~~ corporate paranoia ~~because of~~ state surveillance
output	NSA revelations **strengthen** corporate paranoia **for** state surveillance.
input	~~This seems~~ to ~~be a~~ continuation of the ~~Israeli~~ campaign to ~~prevent~~ the ~~proliferation~~ of ~~weapons~~ in the Middle East.
output	**It appears that this** is the continuation of **Israel's** campaign to **stop** the **spread** of **arms** in the Middle East.

Table 3: Monolingual editing examples from the WMT'14 de-en test set. Strike-through text indicates the rejected words.

Acknowledgments

We thank Marc'Aurelio Ranzato, Sumit Chopra and Roman Novak for helpful discussions. We thank Sergey Edunov, Sam Gross and Myle Ott for writing the fairseq-py toolkit used for these experiments.

References

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

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 597–604.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics* 35(1):3–28.

Steffen Bickel, Peter Haider, and Tobias Scheffer. 2005. Predicting sentences using n-gram language models. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 193–200.

Léon Bottou. 1991. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*. EC2, Nimes, France.

Eric Brill and Robert C Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 286–293.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning .

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

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 *EMNLP*. pages 1225–1236.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. *arXiv preprint arXiv:1709.08878* .

Edgar T. Irons and Frans M. Djorup. 1972. A crt editing system. *Communications of the ACM* 15(1):16–20.

Philipp Koehn. 2009a. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn. 2009b. A web-based interactive computer aided translation tool. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*. Association for Computational Linguistics, pages 17–20.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Transtype: a computer-aided translation typing system. In *Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems-Volume 5*. Association for Computational Linguistics, pages 46–51.

Jindřich Libovickỳ and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. *arXiv preprint arXiv:1704.06567* .

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. volume 1, pages 881–893.

Benjamin Marie and Aurélien Max. 2015. Touch-based pre-post-editing of machine translation output. In *EMNLP*. pages 1040–1045.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady* 27(2).

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 85–91.

Roman Novak, Michael Auli, and David Grangier. 2016. Iterative refinement for machine translation .

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

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* .

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proc. of WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proc. of ACL*.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*. pages 1139–1147.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* .