Enhancing Supervised Learning with Contrastive Markings in Neural Machine Translation Training

Nathaniel Berger[*], **Miriam Exel**[‡]**, Matthias Huck**[‡] and **Stefan Riezler**[†,*]

[*]Computational Linguistics & [†]IWR, Heidelberg University, Germany
[‡]SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany
`{berger, riezler}@cl.uni-heidelberg.de`
`{miriam.exel, matthias.huck}@sap.com`

Abstract

Supervised learning in Neural Machine Translation (NMT) typically follows a teacher forcing paradigm where reference tokens constitute the conditioning context in the model's prediction, instead of its own previous predictions. In order to alleviate this lack of exploration in the space of translations, we present a simple extension of standard maximum likelihood estimation by a contrastive marking objective. The additional training signals are extracted automatically from reference translations by comparing the system hypothesis against the reference, and used for up/down-weighting correct/incorrect tokens. The proposed new training procedure requires one additional translation pass over the training set per epoch, and does not alter the standard inference setup. We show that training with contrastive markings yields improvements on top of supervised learning, and is especially useful when learning from postedits where contrastive markings indicate human error corrections to the original hypotheses. Code is publicly released[1].

1 Introduction

Due to the availability of large parallel data sets for most language pairs, the standard training procedure in Neural Machine Translation (NMT) is supervised learning of a maximum likelihood objective where reference tokens constitute the target history in the conditional language model, instead of the model's own predictions. Feeding back the reference history in model training, known as *teacher forcing* (Williams and Zipser, 1989), encourages the sequence model to stay close to the reference sequence, but prevents the model to learn how to predict conditioned on its own history, which is the actual task at inference time. This lack of exploration in learning has been dubbed *exposure bias* by Ranzato et al. (2016). It has been tackled by techniques that explicitly integrate the model's own prediction history into training, e.g. scheduled sampling (Bengio et al., 2015), minimum risk training (Shen et al., 2016), reinforcement learning (Bahdanau et al., 2017), imitation learning (Lin et al., 2020), or ramp loss (Jehl et al., 2019), amongst others. In most of these approaches, feedback from a human expert is simulated by comparing a system translation against a human reference according to an automatic evaluation metric, and by extracting a sequence- or token-level reward signal from the evaluation score.

In this paper, we present a method to incorporate *contrastive markings* of differences between the model's own predictions and references into the learning objective. Our approach builds on previous work on integrating weak human feedback in form of error markings as supervision signal in NMT training (Kreutzer et al., 2020). This work was conceptualized for reducing human annotation effort in interactive machine translation, however, it can also be used on simulated error markings extracted from an automatic evaluation score. It allows the model to extract a contrastive signal from the reference translation that can be used to re-

[1]`https://www.cl.uni-heidelberg.de/statnlpgroup/contrastive_markings/`

inforce or penalize correct or incorrect tokens in the model's own predictions. Such a reward signal is more fine-grained than a sequence-level reward obtained by a sequence-level automatic evaluation metric, and less noisy than token-based rewards obtained by reward shaping (Ng et al., 1999).

Our hypothesis is that such contrastive markings should be especially useful in learning setups where human postedits are used as reference signals. In such scenarios, contrastive markings are likely to indicate erroneous deviations of machine translations from human error corrections, instead of penalizing correct translations that happen to deviate from independently constructed human reference translations. We confirm this hypothesis by simulating a legacy machine translation system for which human postedits are available by performing knowledge distillation (Kim and Rush, 2016) on the stored legacy machine translations. We define a "legacy" machine translation system as a system which was previously used in production and produced translations for which human feedback was gathered, but which is no longer productive. Knowledge distillation is required because the legacy system is a black-box system that is unavailable to us, but its outputs are available. For comparison, we apply our framework to standard parallel data where reference translations were generated from scratch. Our experimental results show that on both datasets, combining teacher forcing on postedits with learning from error markings, improves results with respect to TER on test data, with larger improvements for the knowledge-distilled model that emulates outputs of the legacy system.

A further novelty of our approach is the true online learning setup where new error markings are computed after every epoch of model training, instead of using constant simulated markings that are pre-computed from fixed machine translation outputs as in previous work (Petrushkov et al., 2018; Grangier and Auli, 2018; Kreutzer et al., 2020). Online error markings can be computed in a light-weight fashion by longest common subsequence calculations. The overhead incurred by the new training procedure is one additional translation pass over the training set, whereas at inference time the system does not require additional information, but can be shown to produce improved translations based on the proposed improved training setup.

2 Related Work

Most approaches to remedy the exposure bias problem simulate a sentence-level reward or cost function from an automatic evaluation metric and incorporate it into a reinforcement- or imitation-learning setup (Ranzato et al., 2016; Shen et al., 2016; Bahdanau et al., 2017; Lin et al., 2020; Jehl et al., 2019; Gu et al., 2019; Xu and Carpuat, 2021).

Methods that are conceptualized to work directly with human postedits integrate the human feedback signal more directly, without the middleman of an automatic evaluation heuristic. The standard learning paradigm is supervised learning where postedits are treated as reference translations (see, for example, Turchi et al. (2017)). Most approaches to learning from error markings adapt the supervised learning objective to learn from correct tokens in partial translations (Marie and Max, 2015; Petrushkov et al., 2018; Domingo et al., 2017; Kreutzer et al., 2020).

The QuickEdit (Grangier and Auli, 2018) approach uses the hypothesis produced by an NMT system and token-level markings as an extra input to an automatic postediting system (APE), and additionally requires markings on the system output at inference time. This requires a dual encoder architecture with the decoder attending to both the source and hypothesis encoders. In this case, convolutional encoders and decoders of Gehring et al. (2017) are used.

Our approach builds upon the work of Petrushkov et al. (2018) and Kreutzer et al. (2020) who incorporate token-level markings as learning signal into NMT training. In contrast to Grangier and Auli (2018), who compute markings offline before training and require them for inference, we only require them during training and calculate markings online. Furthermore, instead of presenting markings to the system as an extra input, they are integrated into the objective function as a weight. While Petrushkov et al. (2018) simulate markings from reference translations by extracting deletion operations from longest common subsequence calculations, Kreutzer et al. (2020) show how to learn from markings solicited from human annotators. In contrast to these approaches, we integrate markings to enhance supervised learning in a true online fashion.

Source	To remove the highlighting , un@@ mark the menu entry .
Hypothesis	Um die Her@@ vor@@ hebung <u>zu entfernen</u> , <u>mark@@ ieren</u> Sie <u>den</u> Menü@@ ein@@ trag .
Reference	Um die Her@@ vor@@ hebung auszu@@ schalten , de@@ aktivieren Sie diesen Menü@@ ein@@ trag .
Markings	1 1 1 1 1 <u>0 0</u> 1 <u>0 0</u> 1 <u>0</u> 1 1 1 1

Table 1: An example of a source, hypothesis, and reference triple along with the contrastive markings generated by comparing the hypothesis to the reference. Markings of 1 indicate a correct subword token, while 0 indicates an incorrect subword token. We used byte-pair encoding (Sennrich et al., 2016) and the "@@" indicate that this token is part of the same word as the following token. We underline and color the incorrect tokens and their corresponding markings red.
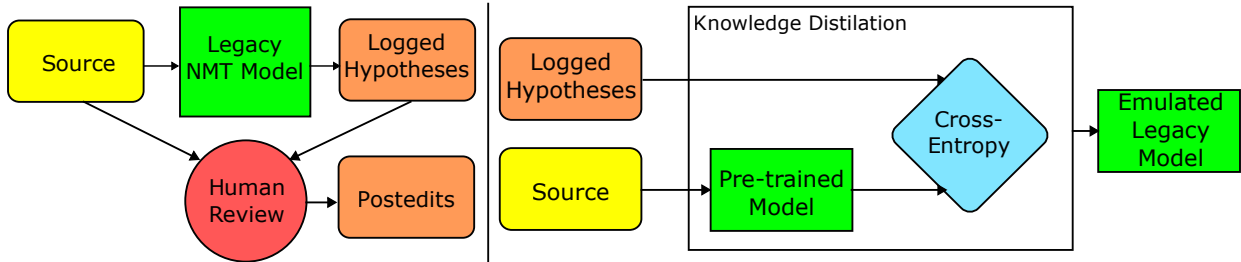
Figure 1: Left: The WMT21 APE dataset is created by having a black-box NMT system generate hypothesis translations. These logged hypotheses are then given to human reviewers to postedit to create a triple of (source, hypothesis, postedit). **Right:** Because the system that generated the hypotheses is not available for us to fine-tune, we try to emulate it with knowledge distillation. We train the model to reproduce the original hypothesis by using them as targets with a cross-entropy loss to produce an emulated legacy model.

3 Methods

3.1 Learning Objectives

Let $x = x_1 \ldots x_S$ be a sequence of indices over a source vocabulary $\mathcal{V}_{\text{SRC}}$, and $y = y_1 \ldots y_T$ a sequence of indices over a target vocabulary $\mathcal{V}_{\text{TRG}}$. The goal of sequence-to-sequence learning is to learn a function for mapping an input sequence x into an output sequence y. For the example of machine translation, y is a translation of x, and a model parameterized by a set of weights θ is optimized to maximize $p_\theta(y \mid x)$. This quantity is further factorized into conditional probabilities over single tokens $p_\theta(y \mid x) = \prod_{t=1}^{T} p_\theta(y_t \mid x; y_{<t})$, where the latter distribution is defined by the neural model's softmax-normalized output vector:

$$p_\theta(y_t \mid x; y_{<t}) = \texttt{softmax}(\text{NN}_\theta(x; y_{<t})). \quad (1)$$

There are various options for building the architecture of the neural model NN_θ, such as recurrent (Bahdanau et al., 2015), convolutional (Gehring et al., 2017) or attention-based (Vaswani et al., 2017) encoder-decoder architectures.

Standard supervised learning from postedits treats a postedited output translation y^* for an input x the same as a human reference translation (Turchi et al., 2017) by maximizing the likelihood of the user-corrected outputs where

$$L_{PE}(\theta) = \sum_{x,y^*} \sum_{t=1}^{T} \log p_\theta(y_t^* \mid x; y_{<t}^*), \quad (2)$$

using stochastic gradient descent techniques (Bottou et al., 2018).

Petrushkov et al. (2018) suggested learning from error markings δ_t^m of tokens t in machine-generated output $\hat{y}$. Denote δ_t^+ if marked as correct, or δ_t^- otherwise, than a model with $\delta_t^+ = 1$ and $\delta_t^- = 0$ will reward correct tokens and ignore incorrect outputs. The objective of the learning system is to maximize the likelihood of the correct parts of the output where

$$L_M(\theta) = \sum_{x,\hat{y}} \sum_{t=1}^{T} \delta_t^m \log p_\theta(\hat{y}_t \mid x; \hat{y}_{<t}). \quad (3)$$

The tokens $\hat{y}_t$ that receive $\delta_t = 1$ are part of the correct output y^*, so the model receives a strong signal how a corrected output should look like. Although the likelihood of the incorrect parts of the sequence does not weigh into the sum, they are contained in the context of the correct parts (in $\hat{y}_{<t}$). Alternatively, it might be beneficial to penalize incorrect tokens, with e.g. $\delta_t^- = -0.5$, and

reward correct tokens $\delta_t^+ = 0.5$, which aligns with the findings of Lam et al. (2019).

Our final combined objective is a linear interpolation of the log-likelihood of postedits L_{PE} and the log-likelihood of markings L_M:

$$L(\theta) = \alpha L_{PE} + (1 - \alpha)L_M. \qquad (4)$$

3.2 Simulating Markings

Error markings are simulated by comparing the hypothesis to the reference and marking the longest common subsequence as correct, as proposed by Petrushkov et al. (2018). We show an example of a data point in Table 1. Markings were extracted from the longest common subsequence calculations. For every token in the model hypothesis there is a corresponding reward. A reward is 0 when the corresponding token is not present in the reference and is 1 when the token was kept in the reference.

3.3 Knowledge Distillation

We want to showcase the advantage of our technique of enhancing supervised learning from human reference translations and from human postedits. In order to take advantage of the fact that human postedits indicate errors in machine translations instead of differences between machine translations and independent human references, we need to simulate the legacy machine translation system that produced the translations that were postedited. For this purpose we use APE data consisting of sources, MT outputs, and postedits. Since the legacy system is a black box to us, we carry out sequence-level knowledge distillation (Kim and Rush, 2016) on the machine translations provided in the train split of the APE dataset (cf. Section 4). This allows us to emulate the legacy system by knowledge distillation and to consider the postedits in the APE dataset as feedback on the knowledge-distilled model. We present an overview of this process in Figure 1.

As shown in Table 2, after fine-tuning on the MT outputs in the train split of the APE data, we are able to produce translations that are more similar to the black-box systems than those of the pre-trained baseline system. Additionally, because the APE dataset's postedits were generated by correcting those MT outputs, Table 3 shows that the knowledge-distilled system's performance on the postedits is closer to the black-box system's performance than before distillation.

3.4 Online Learning

Our learning setup performs standard stochastic gradient descent learning on mini-batches. After every epoch, new system translations are produced and error markings are extracted by comparing the translations to references. This process is shown in Figure 2, showing that we produce error markings by comparing the model's output with the postedits and then use the marked hypotheses and the postedits to train the system.

In preliminary experiments we found that computing error markings from a fixed initial set of system translations and using them as learning signals in iterative training appeared to bring initial improvements. Continued training, however, led to decreased performance. We conjecture that learning from constant marking signals can work for very small datasets (for example, Kreutzer et al. (2020) used fewer than $1,000$ manually created markings for training), but it leads to divergence of parameter estimates on datasets that are one or two orders of magnitude larger, as in this work.

4 Data

We use the WMT17 En-De dataset[2] for pretraining. Our data is pre-processed using the Moses tokenizer and punctuation normalization for both English and German implemented in Sacremoses[3].

We first test our ideas on the IWSLT14 En-De dataset[4] (Cettolo et al., 2012). We download and pre-process the data using joey scripts[5]. The En-De dataset consists of transcribed TED talks and volunteer provided reference translations into the target languages.

The APE dataset is from the WMT automatic postediting shared task 2021 (Akhbardeh et al., 2021). The legacy system that produced the original MT outputs is based on a standard Transformer architecture (Vaswani et al., 2017) and follows the implementation described by Ott et al. (2018). This system was trained on publicly available MT datasets, including Paracrawl (Bañón et al., 2020) and Europarl (Koehn, 2005), totalling 23.7M parallel sentences for English-German. The APE


[2] https://www.statmt.org/wmt17/translation-task.html
[3] https://github.com/alvations/sacremoses
[4] https://sites.google.com/site/iwsltevaluation2014/data-provided
[5] https://github.com/joeynmt/joeynmt/blob/main/scripts/get_iwslt14_bpe.sh

System	Train		Dev		Test	
	BLEU	TER	BLEU	TER	BLEU	TER
APE MT Outputs	100.0	0.0	100.0	0.0	100.0	0.0
Baseline Model	48.0	31.8	49.0	31.0	46.2	33.8
KD Model	88.9	5.8	56.0	25.9	55.8	26.7

Table 2: Systems outputs compared to APE data *MT outputs*. BLEU and TER scores indicate distance of system outputs to MT outputs that were shown to human posteditors. Results show that Knowledge Distillation (KD) on APE MT Outputs improves distances (higher BLEU, lower TER), enabling improved approximation of the MT system that generated the hypotheses used in the APE dataset. Baseline and Knowledge Distillation systems evaluated with a beam size of 5.

System	Train		Dev		Test	
	BLEU	TER	BLEU	TER	BLEU	TER
APE MT Outputs	70.8	18.1	69.1	18.9	71.5	17.9
Baseline Model	42.4	36.9	43.3	35.8	41.7	37.8
KD Model	66.0	20.8	49.1	31.2	49.6	31.6

Table 3: System outputs compared to APE data *postedits*. Results show that Knowledge Distillation (KD) on APE MT outputs also reduces the distance to APE postedits (higher BLEU, lower TER) . Baseline and KD systems are evaluated with a beam size of 5.
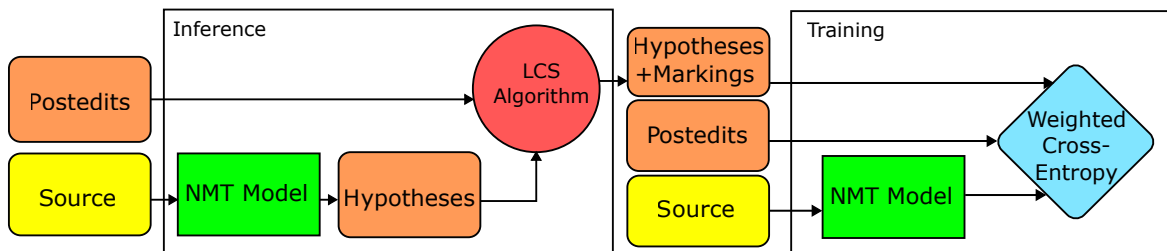
Figure 2: Once per epoch, we have our model run inference on all source sentences to generate hypothesis sentences. These then get compared to the postedits using the Longest Common Subsequence algorithm with tokens contained in the subsequence marked as good and those not in the subsequence marked as bad. Both the marked hypotheses and postedits are used as targets with a weighted cross-entropy loss function. The NMT model that generate the hypotheses and the model we train are the same model.

data consists of source, MT output, and postedit triples. The source data was selected from English Wikipedia articles. The MT outputs were provided by the legacy system and were postedited by professional translators. The sizes of the datasets are given in Table 4.

5 Experiments

5.1 Experimental Setup

We implement our loss function and data-loading on top of JoeyNMT (Kreutzer et al., 2019).[6] All that needs to be changed, in addition to adding weighting to the loss function, is a way of loading data and constructing combined batches such that each batch contains sources, hypotheses, weights, and postedits. To do this, we duplicate each source

twice in the batch and pair the first copy with the hypothesis and the second copy with the postedit. From the point of view of the model and loss function, the batch constructed for the combined objective does not differ from a normal batch with token-level weights. Batches constructed this way and in the usual manner can both contain the same number of tokens, but half of the target sequences in the combined batches come from the model's own translation of the training data.

Our baseline system is a standard Transformer model (Vaswani et al., 2017), pre-trained on WMT17 data for English-to-German translation (Bojar et al., 2017), and available through JoeyNMT[7]. The model uses 6 layers in both the en-

---

[6]https://github.com/joeynmt/joeynmt

[7]https://www.cl.uni-heidelberg.de/statnlpgroup/joeynmt/wmt_ende_transformer.tar.gz

Dataset	Train	Dev	Test
WMT17 (pre-train)	$5,919,142$		
IWSLT14 (fine-tune)	$158,794$	$7,216$	$6,749$
WMT21 APE (fine-tune)	$7,000$	$1,000$	$1,000$

Table 4: Size of En-De datasets used for pre-training and fine-tuning: The WMT17 and IWSLT14 data consist of pairs of source and target sentences; the WMT21 APE data consists of triples of source, MT output, and postedited sentences.

System	References	Online markings	TER
a	1.0	0.0	48.2
b	0.9	0.1	48.1
c	0.7	0.3	$48.0^{a,f}$
d	0.5	0.5	$47.8^{a,f}$
e	0.3	0.7	48.3
f	$\emptyset$	$\emptyset$	51.3

Table 5: Results from fine-tuning the WMT17 News model on out-of-domain IWSLT references. Numbers in the References and Online markings columns refer to interpolation weights given to that loss. The bottom row is the unchanged system, hence its interpolation values are $\emptyset$. The results show that, up to a threshold, increasing the weight given to Online markings improves TER scores. Superscripts denote statistically significant differences to indicated system at p-value < 0.05.

coder and decoder with 8 attention heads each, and hyper-parameters as specified in the pre-trained JoeyNMT model's configuration file.

We compare the combined objective given in Equation (4) to standard supervised fine-tuning by continued training on references or postedits and to the pre-trained model.

All systems share the same hyper-parameters except for the weighting of target tokens. The standard supervised learning method does not account for token-level weights and therefore all weights in the loss-function are set to 1. For the contrastive marking method, we experimented with a range of interpolation values α on the IWSLT14 dataset to select the best value. The weighting of the tokens were set to $-0.5, 0.5$ in correspondence with the results from Kreutzer et al. (2020).

5.2 Experimental Results

Since our work is concerned with learning from token-based feedback, we evaluate all systems according to Translation Edit Rate (TER) (Snover et al., 2006). Furthermore, we provide the Sacre-BLEU (Post, 2018) signatures[8] for evaluation configurations of evaluation metrics. Statistical significance is tested using a paired approximate randomization test (Riezler and Maxwell, 2005).

[8]TER: nrefs:1 | ar:10000 | seed:12345 | case:lc | tok:tercom | norm:no | punct:yes | asian:no | version:2.0.0

Table 5 shows results from fine-tuning on independently created human references. A baseline model trained on WMT17 data (line f) is fine-tuned on references (line a) or on a combination of references and online markings (lines b-e, using different interpolation weights) from the TED talks domain. We see that up to a threshold, increasing the interpolation weight given to learning from online markings significantly improves TER scores up to 3.5 points (line d) compared to the baseline (line f), and up to 0.5 points compared to training from references only (line a).

Table 6 gives an experimental comparison of fine-tuning experiments on human postedits. A baseline model trained on news data is fine-tuned on postedit data from the Wikipedia domain. The postedit data is feedback on real MT outputs that we have trained on using knowledge distillation to emulate. Line a shows TER results for fine-tuning on postedits. This result can be improved significantly by 0.6 TER by combined learning on postedits and online markings, using an interpolation weight of 0.5 (line b). Lines c and d perform the same comparison of objectives for a model that has been trained via knowledge distillation (KD) of the legacy machine translations that were the input data for postediting. Comparing line d to line a, we see that by combined learning of a KD system on postedits and markings even larger gains, close to 1 TER point, can be obtained. The improve-

	System	TER
a	Baseline + Postedits	31.3
b	Baseline + Postedits + Online Markings	30.7^{a}
c	Baseline + KD + Postedits	30.8
d	Baseline + KD + Postedits + Online Markings	30.4^{ac}

Table 6: Fine-tuned systems compared to WMT APE postedit test data. Results show that Online markings, when combined with learning from references, are able to improve our systems more than references alone. Even larger improvements are gained by systems trained by knowledge distillation (KD) on legacy translations. Interpolation weights are set to 0.5. Superscripts indicate a significant improvement $p < 0.05$ over the indicated system.

ments due to adding online markings are significant over training from postedits alone in all cases, and nominally, results for models adapted to the legacy machine translations via KD are better than for unchanged models trained on postedits.

An example showing the learning progress of the different approaches during the first epochs is given in Table 7. The results of epoch 0 are given in the first block. It shows the system outputs of the models trained with knowledge distillation and the baselines before learning from postedits or markings. The KD models, given in lines c and d, already show better terminology translation (superstructure - Überbau, bases - Fundamente) than the baselines in lines a and b (superstructure - Superstruktur, bases - Stützpunkte). After one epoch, contrastive learning (lines b and d) and learning from postedits (lines a and c) correct "armored - gewagelt" and "armored - getrieben" to "armored - gepanzert", but only for KD models or if contrastive learning is used. Furthermore, contrastive learning of a KD model (line d) also corrects the translation of "funnel" from "Funnels" to "Trichter".

6 Discussion

Our experimental results in Table 6 show that online markings combined with references or postedits bring greater improvements than supervised learning on references or postedits alone, and moreover, the knowledge distilled models benefit more from the provided feedback. This suggests that the more related the feedback is to the system's own output, the more can be learned from the feedback.

Furthermore, this result has implications for how to best use postedits. Postedits are often treated as new reference translations for the sources and used to train new systems, whereas the original MT outputs are discarded. However, fine-tuning the original system on the postedits may yield larger improvements than training a new, unrelated model on the source and postedit alone.

Lastly, we believe that our results can be interpreted as the effect of mitigating exposure bias. The pre-trained model is exposed not only to reference translations, but to its own trajectories. Even if the model's trajectory is far from the gold reference and multiple tokens in its history are incorrect, it will be rewarded if it predicts a token that is in the output. This may enable it to return to a more rewarding trajectory.

7 Conclusion

In this work we present a way to combine postedits and word-level error markings extracted from the edit operations between the postedit and the MT output to learn more than what the postedit alone is able to provide. Experimentally, we try this on systems unrelated to the legacy system, whose outputs were originally postedited, and on a simulation of the legacy system we create via knowledge distillation. We show that these contrastive markings are able to bring significant improvements to TER scores and we hypothesize this is because they are able to target insertion errors that contribute to higher TER scores. Additionally, learning from the model's own output may allow it to learn how to correct itself after making an error if it is later rewarded for correct outputs.

References

Akhbardeh, Farhad, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp

Source	the superstructure was armored to protect the bases of the turrets , the funnels and the ventilator ducts in what he termed a breastwork .
Postedit	der Überbau wurde gepanzert , um die Fundamente der Türme , der Trichter und der Ventilatorkanäle in dem Bereich zu schützen , den er als Brustwehr bezeichnete .

Epoch 0

System	Hypothesis
a	die Superstruktur wurde getrieben , um die Stützpunkte der Turm- , der Funn ¤ rn- und der Ventilator die Herde in dem , was er als die Bruststbesteigung bezeichnet hatte zu schützen .
b	die Superstruktur wurde getrieben , um die Stützpunkte der Turm- , der Funn ¤ rn- und der Ventilator die Herde in dem , was er als die Bruststbesteigung bezeichnet hatte zu schützen .
c	der Überbau wurde gewagelt , um die Fundamente der Türme , die Funnels und die Ventilatorenkanäle in einem Brustwerk zu schützen .
d	der Überbau wurde gewagelt , um die Fundamente der Türme , die Funnels und die Ventilatorenkanäle in einem Brustwerk zu schützen .

Epoch 1

System	Hypothesis
a	die Superstruktur wurde gezeichnet , um die Stützen der Turrets , der Funnels und der Ventilator in seiner Art Brustwork zu schützen .
b	die Überbauung war gepanzert , um die Grundstücke der Turrets , der Funnels und der Vaterfunkanten in dem , was er als Brustwerk nannte , zu schützen .
c	der Superbau wurde gepanzert , um die Stützpunkte der Turrets , der Funnels und der Ventilatorentötungen in einer so genannten Brustarbeit zu schützen .
d	der Überbau wurde gepanzert , um die Fundamente der Türme , der Trichter und der Ventilatorkankanäle zu schützen , was er als Brustwerk nannte .

Table 7: Here we show the beginning of a training trajectory for a single example from the APE dataset. Above is the source and the postedit from the dataset, after which follows the first three epochs. Because translations and markings are generated before the beginning of an epoch, epoch 0 contains outputs from the knowledge distilled (KD) (lines c and d) and baseline systems (lines a and b). The systems letters correspond to those in Table 6, indicating learning from postedits in lines a and c, and learning additionally from the contrastive markings in lines b and d. Models c and d have seen the MT side of this dataset beforehand and are already more capable of translating terminology such as "superstructure" to "Überbau". After one epoch, we see that the KD models and the contrastive learning objective models are able to correct "gewagelt" and "getrieben" to "gepanzert" as the translation of "armored". Because we use subword tokens, we have markings on portions of words. Although "Überbau" is a part of "Überbauung", the subwords used to construct them differ, leading to "bau" in "Überbauung" being marked as incorrect.

Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online, November. Association for Computational Linguistics.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA.

Bahdanau, Dzmitry, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Bañón, Marta, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis,

Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online, July. Association for Computational Linguistics.

Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the*

Second Conference on Machine Translation, Volume 2: Shared Task Papers, pages 169–214, Copenhagen, Denmark, September. Association for Computational Linguistics.

Bottou, Leon, Frank E. Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311.

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy, May 28–30. European Association for Machine Translation.

Domingo, Miguel, Álvaro Peris, and Francisco Casacuberta. 2017. Segment-based interactive-predictive machine translation. *Machine Translation*, 31(4):163–185.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

Grangier, David 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, June. Association for Computational Linguistics.

Gu, Jiatao, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32.

Jehl, Laura, Carolin Lawrence, and Stefan Riezler. 2019. Learning neural sequence-to-sequence models from weak feedback with bipolar ramp loss. *Transactions of the Association for Computational Linguistics*, 7:233–248.

Kim, Yoon and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas.

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

Kreutzer, Julia, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China, November. Association for Computational Linguistics.

Kreutzer, Julia, Nathaniel Berger, and Stefan Riezler. 2020. Correct me if you can: Learning from error corrections and markings. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 135–144, Lisboa, Portugal, November. European Association for Machine Translation.

Lam, Tsz Kin, Shigehiko Schamoni, and Stefan Riezler. 2019. Interactive-predictive neural machine translation through reinforcement and imitation. In *Proceedings of the Machine Translation Summit (MT-SUMMIT XVII)*, Dublin, Ireland.

Lin, Alexander, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Marie, Benjamin and Aurélien Max. 2015. Touch-based pre-post-editing of machine translation output. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.

Ng, Andrew Y., Daishi Harada, and Stuart J. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, Bled, Slovenia.

Ott, Myle, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October. Association for Computational Linguistics.

Petrushkov, Pavel, Shahram Khadivi, and Evgeny Matusov. 2018. Learning from chunk-based feedback in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representation (ICLR)*, San Juan, Puerto Rico.

Riezler, Stefan and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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

Shen, Shiqi, Yong Cheng, Zongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.

Snover, Matthew, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12. Association for Machine Translation in the Americas.

Turchi, Marco, Matteo Negri, M. Amin Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 1(108):233–244.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Williams, Ronald J. and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Xu, Weijia and Marine Carpuat. 2021. EDITOR: An edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.