



End-to-end entity-aware neural machine translation

Shufang Xie¹ · Yingce Xia² · Lijun Wu² · Yiqing Huang³ · Yang Fan⁴ · Tao Qin²

Received: 13 May 2021 / Revised: 13 August 2021 / Accepted: 22 September 2021 /

Published online: 13 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2021

Abstract

Accurate translation of entities (e.g., person names, organizations, geography) is important in neural machine translation (briefly, NMT), as they are usually more difficult to translate than other words, and an incorrect translation of them will greatly hurt user experiences. In previous works, entities are either treated in the same way as other words, which leads to inaccurate translation, or handled by multiple steps (including named entity recognition, translation, and replacing entities back), which significantly increase the inference latency. In this work, we propose an end-to-end algorithm that carefully handles the translation of entities. There are mainly two novel parts compared to conventional NMT model: (1) The encoder and the decoder are attached with entity classifiers, which are used to verify whether the input token is a named entity. In this way, the encoder and decoder are capable to treat named entities differently; (2) The translation loss of each target token is adaptively increased by the probability that the target token is a named entity, which results in more accurate translation of entities. During inference time, these two parts will be removed so that the translation model maintains the same inference speed as conventional NMT models. Empirical results on six translation tasks demonstrate the effectiveness of our methods of improving the translation quality. Specifically, we improve 1.7 BLEU scores on Japanese to English translation and 4.6 entity F_1 scores on English to Chinese translation, without additional inference cost.

Keywords Machine translation · Named entity

1 Introduction

Neural machine translation (briefly, NMT), which is to translate a sentence from the source language to target language with deep neural networks, has made great progress in recent years (Wu et al. 2016; Hassan et al. 2018; Luong et al. 2016). Despite the success of previous works, most of them only focus on improving the general translation quality, where each word makes the same contribution to the evaluation metrics.

Editors: Yu-Feng Li, Mehmet Gönen, Kee-Eung Kim.

✉ Shufang Xie
shufangxie@ruc.edu.cn

Extended author information available on the last page of the article

However, words are not all equally important in a sentence and can have different impacts on the translation quality for human evaluation. Intuitively, *entities* are likely to contain critical information in the sentence and are more important for translation quality (Li et al. 2018; Post et al. 2019; Niehues and Cho 2017), where detailed statistics are available in Sect. 5.1. Consider two translation systems A and B and a source sentence with ground-truth translation “Both Alice and Bob live in *Washington State*.” as a simple example. Suppose the translation result of system A is “Both Alice and Bob live in *Wisconsin State*.”, and that of system B is “Both Alice and Bob live *at Washington State*.”. Although both systems make a mistake on single word, clearly, we vote system B as a better one, since it correctly conveys the important information about the location. That is, the named entities (e.g., locations, numbers) are important for user experience.

Therefore, in this work, we focus on improving the translation quality of named entities. Unfortunately, entities are not easily translated. Previous studies (Hassan et al. 2018) have shown that today’s NMT systems do not perform very well for entity translation.

In previous work, the general solution is adding extra entity information to the NMT system during both training and inference time. Multiple steps are required for this solution: first, we need to detect entities in both source and target sentences using a named entity recognition (briefly, NER) tool; second, the entities should be tagged, like replaced with special placeholders (Wang et al. 2017; Post et al. 2019; Li et al. 2018), adding special tokens to indicate the boundaries (Li et al. 2018; Modrzejewski et al. 2020), using code-switching method Song et al. (2019), or labeled with entity embedding to enhance the translation (Sennrich and Haddow 2016; Niehues and Cho 2017; Ugawa et al. 2018); finally, NMT models are trained on the processed sentences. During the inference time, the input should be processed using NER tools, translated to the target language, and post-processed (e.g., replacing placeholders back, removing extra tags) to get the translation.

The accuracy of recognizing the named entities greatly affects the translation quality. As the development of pre-training (Devlin et al. 2019; Liu et al. 2019), the NER tools are significantly improved (Burtsev et al. 2018; Luo et al. 2020). However, those NER modules built upon pre-trained models are even heavier than NMT models. For instance, the numbers of parameters of DeepPavlov (Burtsev et al. 2018), one of the state-of-the-art models for NER, is more than ten times of Transformer used NMT in industry (Kim et al. 2019). The reason is that the DeepPavlov models are based on the large scale pre-training. As a result, it is not feasible to directly integrate such a heavy NER module during the inference time due to a large amount of overhead.

In this work, we design an end-to-end entity-aware NMT model, where both the encoder and decoder can serve as named entity recognizers but there is no extra cost at inference time. During training time, similar to aforementioned works, we leverage a NER tool to provide entity tags for the source and target sentences in the training corpus. When training the translation models, in addition to translation loss, we add NER detection loss to both the encoder and decoder, so that they can correctly recognize the entities tagged by the NER tool. Furthermore, to pay more attention to named entities and differentiate them from the other words, we assign the entities in the target sequence with larger weights inspired by the focal loss (Lin et al. 2017). In this way, the NER module and the translation network are closely coupled and collaborate through end-to-end training, boosting the performance of both tasks. The inference process is the same as that for standard NMT, which does not invoke additional costs. This allows us to use arbitrary high quality NER model during the training, without hurting the inference efficiency.

To summarize, the main contributions of this work are three-fold:

- We introduce a novel end-to-end method to improve the translation quality, especially for the accurate translation of named entities in sentences.
- Compared with previous methods, we keep the one-pass decoding process without the dependency of heavy NER model for inference.
- Experiments on six translation tasks extensively verify the effectiveness of our method. According to the results, our method can improve both BLEU score and entity F_1 score.

2 Related work

Enhancing NMT systems with knowledge is a promising research direction in recent years. For example, in Lu et al. (2018), Zhao et al. (2020), Zhao et al. (2020), knowledge graph is incorporate into machine translation task, and in Zhu et al. (2020), Clinchant et al. (2019), Yang et al. (2020), Shavarani and Sarkar (2021), pre-trained language models are used to enhance the translation. In this work, named entity information is leveraged to boost the performance. Named entity is an important topic in the NLP area and there are many previous work to improve the entity translation quality. The common approach is introducing the entity information to the NMT systems. And then translation models can handle the entity in input sentences better with the help of such information. In previous work, there are different approaches to make use of the entity information. The details are listed as follows:

Placeholder In this kind of methods, entities in the source sentences are masked by placeholders. Wang et al. (2017) use the $\$TERM$ token to mask person name. And Post et al. (2019) mask various entity tokens like numbers, names, cities, emoji, etc. In Li et al. (2018), entities are masked by the type and index, e.g. $LOC1$, $LOC2$, etc. After translation, the masks will be replaced back in target languages, either by entity index or alignment.

Special tokens In Li et al. (2018), Modrzejewski et al. (2020), special tokens are used to indicate the beginning and end of entities in source sentences. For example, the “Hyrule” in a source sentence will become “ $\langle LOC \rangle Hyrule \langle / LOC \rangle$ ” after preprocess to indicate that the word is a location. After translation, the extra tokens will be removed from the model output.

Code-switching In Song et al. (2019), the authors use a code-switching method on entity words. The source side entities are replaced by the corresponding translation in target language. After such preprocess, the input to the model is a combination of source and target language. Therefore, the NMT models only need to copy the those tokens.

Entity embedding The embedding based methods are important direction in previous work. In Sennrich and Haddow (2016), Niehues and Cho (2017), linguistic input features are used to improve model quality. And in Ugawa et al. (2018), the entity embedding is added to token embedding to enhance the representation of sentences.

Despite the success of previous work, the complexity of those methods is still a obstacle for them to be used in real scenario. Especially, the extra cost of NER is not negligible and will significantly affect the decoding latency. Compared with the existing methods, our system has almost zero extra cost during the inference time and better performance.

3 Our methods

In this section, we first introduce the notations, and then describe the network architecture in Sect. 3.1 and the training strategy is in Sect. 3.2.

Notations Let $X = (X_0, X_1, \dots, X_{M-1})$ denote a source sequence with length M , and let $Y = (Y_0, Y_1, \dots, Y_{N-1})$ denote the corresponding target sequence with length N . X_i and Y_i represent the i -th token in X and Y , which can be words or subwords (Sennrich et al. 2016) in natural language. Let X^{ne} and Y^{ne} denote the entity sequences for X and Y . X_i^{ne} and Y_i^{ne} are the named entity tags for X_i and Y_i respectively. The entities are represented as IOB tagging, where “I” represents the inside and is extended to the end of an entity, “O” means that the token is outside of entity, and “B” stands for the beginning of an entity,

Following multilingual version of DeepPavlov NER model¹, we have 19 different kinds of entities in total, which constructs a set \mathbb{N} . There is a special token \circ in \mathbb{N} , which represents that the token is not a named entity. The full list of the supported entity types can be found at Appendix.

3.1 Network architecture

We use Transformer (Vaswani et al. 2017) as the backbone of our model, where the encoder and decoder are modified to be an entity-enhanced version. However, our technique can be easily integrated into other encoder-decoder based models as well. The network architecture is shown in Fig. 1.

Entity-enhanced encoder Let enc denote the encoder of the standard Transformer made up of several stacked blocks. Each block consists of a self-attention layer and a feed-forward layer. Given the input X , enc processes it into hidden representations, which is mathematically defined as $H^{\text{src}} = \text{enc}(X)$. H^{src} is the output of the last block in enc , regarded as a $M \times d$ matrix, where the i -th row H_i^{src} denotes the representation of token X_i , and the d means the embedding dimension.

After that, the encoder works as follows:

$$\begin{aligned} H^{\text{ne}} &= \text{ReLU}(H^{\text{src}} W_{\text{ne}}^{\text{src}}), \\ H^{\text{enc}} &= H^{\text{src}} + H^{\text{ne}}, \\ \hat{X}^{\text{ne}} &= \text{softmax}(H^{\text{ne}} E^{\text{s-ne}}), \end{aligned} \quad (1)$$

where $W_{\text{ne}}^{\text{src}}$ is a $d \times d$ matrix to be learned, $E^{\text{s-ne}}$ is the entity embedding of the source language with size $d \times |\mathbb{N}|$, and \hat{X}^{ne} means the matrix of the predicted entity tokens of X . \hat{X}^{ne} is only required during training, and we do not need it at inference time.

In Eqn.(1), the representation H^{src} is fed into a feed-forward layer and get H^{ne} . After applying an affine transformation to H^{ne} and a softmax operation, we can get the predicted entities \hat{X}^{ne} of the input sequence X . We will minimize the difference between \hat{X}^{ne} and X^{ne} (i.e., outputted by the NER model) so that H^{ne} can be regarded as the features of the entities. We add H^{src} and H^{ne} together as the eventual output of the encoder and feed it into the decoder. In this way, both the named entity information and the semantic information represented by natural words can be passed into the decoder.

Entity-enhanced decoder Similarly, we define dec as the decoder of the standard Transformer, which is also made up of a series of blocks. Beside a self-attention and a feed-forward layer, each block also consists of an additional encoder-decoder attention layer, which is used to aggregate the information from the encoder, i.e., H^{enc} . Let

¹ https://github.com/deepmipt/DeepPavlov/blob/0.10.0/deeppavlov/configs/ner/ner_ontonotes_bert_mult.json

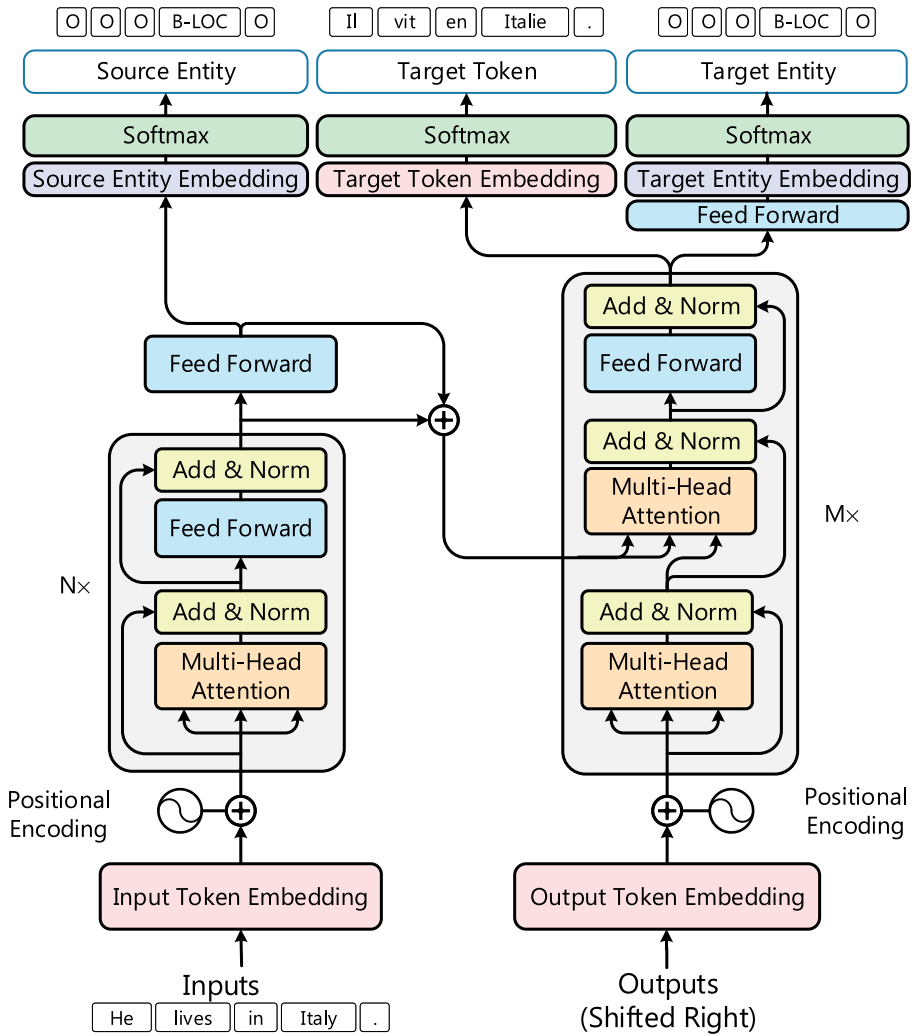


Fig. 1 Network architecture

$Y_{<t}$ denote the sub-sequence $(Y_0, Y_1, \dots, Y_{t-1})$, where Y_0 is a special token indicating the beginning of a sentence. The decoder works as follows:

$$\begin{aligned}
 H_t^{tgt} &= \text{dec}(H^{\text{enc}}, Y_{<t}), \\
 \hat{Y}_t^{\text{ne}} &= \text{softmax}(\text{ReLU}(H_t^{tgt} W_{\text{ne}}^{\text{tgt}}) E^{\text{t-ne}}), \\
 \hat{Y}_t &= \text{softmax}(H_t^{tgt} E^{\text{t}}),
 \end{aligned}
 \tag{2}$$

where $E^{\text{t-ne}}$ is the entity embedding of the target language, and E^{t} is the embedding of target words. The $W_{\text{ne}}^{\text{tgt}}$ is a $d \times d$ affine matrix. Specifically, we first get the representation of the last block in the decoder and transform it into entity embedding space, and then use

different softmax operations to get the predicted translated token and the predicted entity tag, respectively.

During the inference time, we can skip the entity tag predictions and only keep the token prediction, which leads to similar decoding cost as standard MT methods.

Discussion The key challenge in this task is that there are no entity labels available during the inference time. To solve this, we leverage a multi-task framework to build the enhanced encoder and decoder, where the primary task is machine translation, and two auxiliary tasks are source-side and target-side named entity detection. Previous work Niehues and Cho (2017) shows that multi-task learning can help improve the performance. Therefore, we can not only improve the accuracy on named entities, but also regularize the training. As for the NER classification, we can also leverage the outputs of internal blocks of the encoder and decoder. We empirically verified the effect on the choices of these different outputs and found that there are no significant differences compared with using the output from last block.

3.2 Training and inference strategies

Let θ denote the parameters of enc, dec and word embeddings. Let $\theta_{s\text{-ne}}$ and $\theta_{t\text{-ne}}$ denote the parameters related to source-side NER and target-side NER.

The training loss consists of the following three parts:

$$\begin{aligned}\ell_{s\text{-ne}} &= -\frac{1}{M} \sum_{i=0}^{M-1} \log P(X_i^{\text{ne}} | X; \theta, \theta_{s\text{-ne}}), \\ \ell_{t\text{-ne}} &= -\frac{1}{N} \sum_{j=0}^{N-1} \log P(Y_j^{\text{ne}} | X, Y_{<j}; \theta, \theta_{s\text{-ne}}, \theta_{t\text{-ne}}), \\ \ell_{\text{mt}} &= -\frac{1}{N} \sum_{t=0}^{N-1} (1 + P_{\text{NE},t}^{\gamma}) \log P(Y_t | Y_{<t}, X; \theta, \theta_{s\text{-ne}}), \\ P_{\text{NE},t} &= P(Y_t^{\text{ne}} \neq 0 | X, Y_{<t}; \theta, \theta_{s\text{-ne}}, \theta_{t\text{-ne}}),\end{aligned}\tag{3}$$

where there are two losses for named entity recognition $\ell_{s\text{-ne}}$, $\ell_{t\text{-ne}}$, and a translation loss ℓ_{mt} . Because the human annotations of source and target entity labels are unavailable, the labels extracted by DeepPavlov are used to compute the entity loss for both $\ell_{s\text{-ne}}$ and $\ell_{t\text{-ne}}$. For the translation loss ℓ_{mt} , considering that we should enhance the entity tokens, we design an adaptive way inspired from the focal loss (Lin et al. 2017): $P_{\text{NE},t}$ is the probability that Y_t is an entity token (instead of 0). The more likely the token is an entity, the larger weight we will assign to it. Following Lin et al. (2017), the weight is controlled by a positive hyper-parameter γ for flexibility. The weight of each token is at least one to stabilize training.

The final training objective function of entity-enhanced NMT model on data pair (X, Y) is

$$\ell = \ell_{\text{mt}} + \alpha \ell_{s\text{-ne}} + \beta \ell_{t\text{-ne}},\tag{4}$$

where α and β are hyper-parameters to be tuned according to validation performance. Practically, the hyper-parameter setting in all our experiments are: $\gamma = 1.0$, $\alpha = 0.5$, $\beta = 0.5$.

Table 1 Dataset statistics

Language pair	En↔ De	En ↔ Zh	En ↔ Ja
# Train sentence	160k	234k	3.9M
# Train SRC entity	429k	637k	8.6M
# Train TGT entity	439k	571k	8.8M
# Dev sentence	7k	4k	4k
# Dev Src entity	20k	10k	18k
# Dev Tgt entity	20k	9k	17k
# Test sentence	6.8k	1.5k	4k
# Test Src entity	16k	4k	18k
# Test tgt entity	16k	3k	16k
Subword operation	10k	10k	16k
Joint vocab	yes	no	no

The “Train”, “Dev” and “Test” represent training, validation, and test sets. For En ↔ Ja, the test set size for KFTT, JESC, and TED are 1k, 2k and 1k, respectively. The entities are count after subword operation

Table 2 Example of subword entity tags assignment

Sentence	Jon	Lives	In	Winterfell	.
NER Token	Jon	lives	in	Winter	fell .
NER Tag	B-PER	O	O	B-LOC	I-LOC O
BPE Token	Jon	lives	in	Win@@	ter@@ fell .
Aligned Tag	B-PER	O	O	B-LOC	I-LOC I-LOC O

At inference time, we are interested in the translation, therefore we will ignore the related named entity recognition modules. Specifically, the aforementioned \hat{X}_t^{ne} and \hat{Y}_t^{ne} only affect the training process, our method therefore maintains efficiency in inference.

The NER module in the decoder is also disabled and we only generate the translation sentence.

4 Experiments

Data processing We conduct experiments on the translation of four languages, English, German, Chinese, and Japanese, which are briefly denoted as En, De, Zh and Ja respectively. It includes both linguistic distance close language pairs En↔De and more different languages like En↔Zh, En↔Ja. We follow Ott et al. (2019) to process data for IWSLT’14 En↔De, where all words are lowercased and tokenized. We follow Zhu et al. (2020) to process the data for IWSLT’17 En↔Zh. For En↔Ja, we follow Michel and Neubig (2018), Wang et al. (2019) to combine the training sets of KFTT, JESC, and TED talks together (Neubig 2011; Pryzant et al. 2018; Cettolo et al. 2012), and test on the corresponding test sets separately. For En↔Zh, we use Moses and Jieba tokenizer respectively, after which we use BPE to split them into subwords. For Ja, we use SentencePiece to process it directly. Detail information are in Table 1 and URLs of data and tools are in Appendix.

Practically, the tokenizer leveraged by the NER tool is different from that in NMT pre-processing. To solve this problem, we leverage the fact that tokenization will only affect the non-space characters. We therefore align the entity tags with NMT data by character overlap and adjust the IOB notations accordingly. An example is shown in Table 2, where the location entity word “Winterfell” is split into three parts by BPE then assigned tags accordingly.

Entity-rich test set To better evaluate the performance of our methods, we build two extra entity rich test sets: (1) ER-IWSLT, where ER is short for entity rich. for En→De, we concatenate the test sets of IWSLT from year 2010 to 2017, IWSLT-10 validation sets as a larger one. (2) ER-WMT: for En→De, we concatenate the test sets of WMT from 2014 to 2019. Then, we filter the sentences from them with different thresholds of the number of entities per sentence in the English side and report the corresponding scores.

Configuration The backbone of models consists of six layers in both encoder and decoder. In `transformer_small` configuration, the feed-forward layer dimensions and dropout rate are 256, 1024 and 0.3, and in `transformer_base` setting, they are 512, 2048, 0.1, respectively. Following Vaswani et al. (2017), all models are trained with learning rate 5×10^{-4} by Adam optimizer Kingma and Ba (2015) with `invert_sqrt` learning rate scheduler (Vaswani et al. 2017) and 4096 tokens per GPU. The `transformer_small` models are trained on single P40 GPU while `transformer_base` models are trained on 4 P40 GPUs.

Evaluation We evaluate both translation quality and entity accuracy. For En ↔ De, we use `multi-bleu.perl` script² to evaluate the translation BLEU score for fair comparison with previous works. For other language pairs, we use `sacreBLEU` (Post 2018). We use beam size with 5 and length penalty 1.0 for all language pairs. For entity accuracy, we choose the entity F_1 score as the metric. We use DeepPavlov NER model to extract the entities of both reference and translation files, and then calculate the F_1 score between them by exactly matching. To avoid the bias of DeepPavlov, we also measure the entity quality by Stanford NER tagger (Finkel et al. 2005) as well.

5 Results and discussions

This section is organized as follows: First, we show the relation between entity translation quality and human evaluation, which indicates the importance of entity translation. Then we show the model performance on various language pairs and data sets, as well as the case study to better demonstrate the effects. Finally, we have comprehensive study on the effect of entity types, model configuration, NER tools, and decoding loss.

5.1 Entity and human evaluation

We firstly study how entity translation quality affects human evaluation. We collect 70 translation submissions of 5 languages pairs from WMT19 website³, which contains 131k sentences. The corresponding official human evaluation scores from the WMT19 machine translation challenge report (Bojar et al. 2017) are also collected to calculate the Pearson

² <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>

³ <http://data.statmt.org/wmt19/translation-task/wmt19-submitted-data-v3.tgz>

Table 3 Correlation between entity F_1 score and human evaluation

Language pair		# Systems	Correlation
English	→ German	22	96.41
Chinese	→ English	15	89.28
English	→ Czech	11	87.76
English	→ Russian	12	96.29
Lithuanian	→ English	11	94.95

correlation coefficient between the entity quality (in terms of F_1 score) and the human evaluation results which are shown in Table 3. The results indicate that the entity translation quality measured by DeepPavlov is consistent with the human judgment of the sentence quality.

5.2 Translation quality on normal test sets

The results of 10 normal test sets are listed in Tables 4 and 5. For $En \leftrightarrow \{De, Zh\}$ we compare our system with standard Transformer and other entity placeholder-based methods Post et al. (2019), where the entities in data are replaced with special token indicating the entity type and index (e.g. “⟨PER-0⟩”, “⟨PER-1⟩”, etc.). To simulate the inference process of these methods, we first build an entity mapping table from the training data for each language pair with DeepPavlov and Fast Align Dyer et al. (2013), then replace the entity back by encoder-decoder attention (denote as PH_Align) and entity index (denote as PH_Index). We also compare our method with code-switch method Song et al. (2019) and the entity tagging method Li et al. (2018). For $En \leftrightarrow Ja$, due to the computation resource we only compare with LSTM and standard Transformer.

From these tables, we can see that our models enjoy improvements for both BLEU score and entity F_1 score with the help of end-to-end training of both NMT and NER tasks. Compared with standard Transformer, the entity F_1 score is improved from 0.7 point to 4.6 points on various test sets. For BLEU score, we achieve at most 1.7 point improvement on JESC $En \rightarrow Ja$. On $En \leftrightarrow \{De, Zh\}$ data, we additionally use the paired bootstrap resampling method Koehn (2004) for testing the statistical significance and report the p-value of BLEU score by comparing our system and the Transformer baseline system.⁴ The results suggest that the improvements are statistically significant. The best p-value is 0.001 and the worst is about 0.1. We also report the BLEU scores of Transformer from previous works, which show that our reproduction of the baseline system is comparable or stronger than before. Another finding is that the placeholder-based methods will hurt both BLEU and entity translation performance. The index-based replacement is usually better than alignment but still worse than the baseline. We suspect the reason is the difficulty of obtaining high quality entity translation pairs without large amount of human effort.

Compared with the entity tag method, our system yields similar or even better results, under the condition that we lost help on DeepPavlov NER model during the inference time. We also record the relative latency of adopting our method and other methods against the

⁴ <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl>

Table 4 Experiment results backbone on transformer_small on En ↔ {Zh, De}

System	Δ Latency	#Param	En → Zh		Zh → En		En → De		De → En		
			F_1	BLEU	F_1	BLEU	F_1	BLEU	F_1	BLEU	
Transformer Zhu et al. (2020)	–	–	–	26.3	–	20.1	–	–	28.6	–	34.6
Transformer*	0 ms	14.1 M	37.95	26.1	45.56	20.4	51.13	–	29.0	61.43	35.0
PH_Align Post et al. (2019)	86.7 ms	547.5 M	15.11	23.8	18.61	18.1	16.09	–	27.3	24.54	32.4
PH_Index Post et al. (2019)	86.7 ms	–	18.00	24.1	23.13	18.4	18.03	–	27.4	26.69	32.6
Entity Tag Li et al. (2018)	83.9 ms	–	40.02	26.0	47.53	20.5	52.60	–	29.1	63.54	35.4
Code Switch Song et al. (2019)	89.0 ms	–	38.75	25.9	25.34	16.9	48.25	–	28.9	57.83	34.3
Ours	4.4 ms	15.2 M	42.52	26.4♣	48.86	21.3♣	53.08	–	29.4♠	63.34	35.4♠

The first group is standard Transformer, the second group contains models with entity information, and the last row is our method. * denotes our own re-implementation. Suit symbols denote the significance level (p value) : ♣ 0.1, ⋄ 0.01, ♠ 0.001

Table 5 Experiment results backbone on `transformer_base` on TED, KFTT and JESC En ↔ Ja test sets.

Direction	System	Latency	# Param	TED		KFTT		JESC	
				F_1	BLEU	F_1	BLEU	F_1	BLEU
En → Ja	LSTM Michel and Neubig (2018)	–	–	–	14.5	–	20.8	–	15.8
	Transformer*	252.1 ms	75.7 M	33.90	18.6	46.42	26.7	49.38	24.1
	Ours	256.1 ms	80.9 M	34.59	18.6	47.35	26.9	51.41	24.1
Ja → En	LSTM Michel and Neubig (2018)	–	–	–	13.3	–	20.8	–	18.0
	Transformer Wang et al. (2019)	–	–	–	16.2	–	23.6	–	16.1
	Transformer*	250.2 ms	75.7 M	45.83	18.1	45.60	23.8	47.88	23.2
	Ours	255.3 ms	80.9 M	46.63	19.2	46.35	24.3	48.79	24.9

* denotes our own re-implementation

Table 6 Compare with previous works on De → En

System	Entity F_1	BLEU
Transformer Zhu et al. (2020)	–	34.6
LightConv Wu et al. (2019)	62.18	34.9
DynamicConv Wu et al. (2019)	62.86	35.3
Joint Attention Transformer Fonollosa et al. (2019)	62.25	35.7
Ours	63.34	35.4

Bold values indicate statistically significant $p < 0.01$

baseline Transformer as well as the number of model parameters for further comparison. It can be witnessed that our model only yields 5.24% latency and contains 2.78% parameter compared with the entity tag method. The additional cost is almost negligible when compared with the standard Transformer. Such results indicate that our end-to-end method, which saves both time and memory, is more appropriate in the practical implementation of NMT systems.

Moreover, we compare our method with previous non-entity methods for De → En in Table 6, like Joint Attention Transformer model Fonollosa et al. (2019), LightConv, and DynamicConv Wu et al. (2019). Even though they can also improve the BLEU score, the entity accuracy scores are all below our method. It tells that simply improving the general translation quality cannot guarantee the improvement of entity translation quality.

5.3 Translation quality on entity-rich test sets

To further assess the ability of entity translation of our method, we also test our system on the entity-rich test sets that are described in Sect. 4. The evaluation results are shown in Fig. 2, where the x -axes represent the least number of entities in a sentence, and y -axes denote the BLEU score in left graph and entity F_1 score in right graph. The dash lines represent the results from baseline systems and the solids lines are from our methods. We use

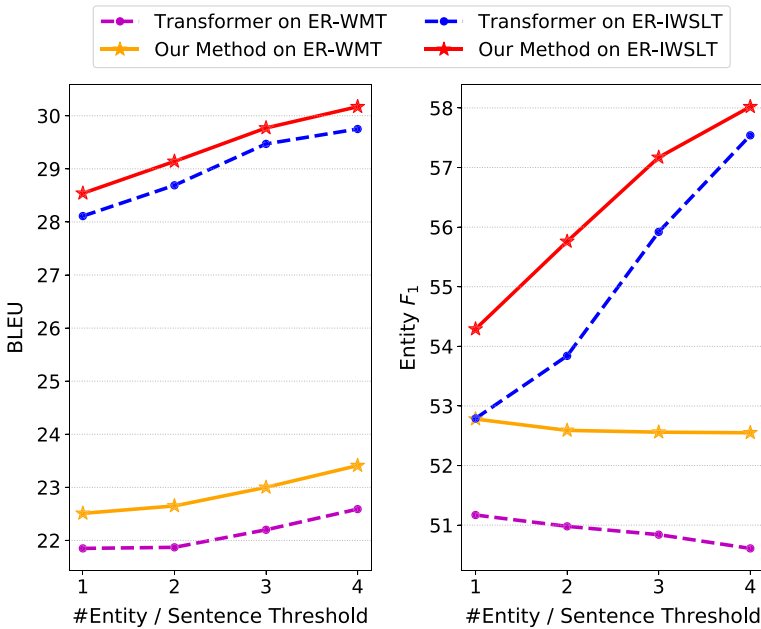


Fig. 2 Translation results on entity-rich test sets. The values on horizontal axes are the thresholds of the number of entities per sentence. The scores on vertical axes are corpus BLEU (left graph) or entity F_1 (right graph) for selected sentences

different colors to distinguish the models and test sets, i.e., orange and red for our method on ER-WMT and ER-IWSLT respectively, purple and blue denotes the baseline Transformer on these datasets. Our model consistently outperforms the baseline models in terms of both BLEU and F_1 . While the WMT test set is in the news domain, which is different from that of the training data, our method can make improvements to the cross-domain translation compared to the baseline Transformer system. It demonstrates that our methods can generalize better across different domains.

5.4 Case study

To emphasize how our method improves the quality of entities, we also conduct translation quality human evaluations with source-based direct assessment (DA) method Bojar et al. (2017) based on Zh \rightarrow En test set. The detailed translation quality human evaluations are attached in the Appendix. The quality of our results outperforms Transformer by 2.1% in terms of average score judged by human annotators. Some of the cases are illustrated in Table 7.

The “Src” and “Ref” represent the source and reference sentences. The H_{TR} , H_{ET} denote the hypotheses generated by standard Transformer and entity tag system. Our results are shown in rows starting with “Ours”. Considering that the entity quality serves as a useful indicator for human evaluation suggested in Table 3, promoting the entity quality as our method does is an appropriate way to enhance the user experience in the practical

Table 7 Examples of Zh → En translation

Src	著名的治疗师 <u>M.斯科特·派克</u> 说过，真正的倾听需要把自己放在一边。
Ref	The famed therapist <u>M. Scott Peck</u> said that true listening requires a setting aside of oneself.
H _{TR}	<u>Scott Papuk</u> : The great therapist, the famous therapist, has to put himself on the side.
H _{ET}	The famous therapist, <u>MM Papker</u> , said that true listening needs to be placed on one side.
Ours	And the famous therapist <u>M. Scott Peck</u> said that the real listening needs to be put on the side.
Src	它外围的礁石形成了保护屏障 中间的环礁湖成为了加拉帕戈斯鲨鱼的幼儿园。
Ref	Its reef forms a protective outer barrier and the inner lagoon is a nursery ground for <u>Galapagos</u> sharks.
H _{TR}	It's a reef around the outside that's formed to protect the middle of the barrier to become kindergarten shark shark.
H _{ET}	It's a reef out of its periphery that formed a nursery to protect the middle of the barrier, which became a kindergarten for sharks in the <u>Galapagos</u> .
Ours	It's surrounded by reefs that form protective lakes in the middle of the barrier that became kindergarten by the <u>Galapagos</u> shark.
Src	数十年的过度捕捞一度将这个村子带到破产边缘
Ref	<u>Decades</u> of overfishing had brought them close to collapse.
H _{TR}	<u>decades</u> of overfishing has brought this village to the brink of bankrupt.
H _{ET}	<u>For ecades, decades</u> of overfishing led this village to bankruptcy.
Ours	For <u>decades</u> , overfishing had taken this village to the edge of the bankrupt.
Src	苹果中90%的养分 — 苹果中所有的抗氧化剂 — 在我们购买时全部流失完了。
Ref	<u>Ninety percent</u> of the quality of that apple – all of the antioxidants – are gone by the time we get it.
H _{TR}	<u>Ninety percent</u> of all of the nutrients in the apple – all the antioxidants in the apple – lost everything when we bought it.
H _{ET}	<u>Ninety-nine percent</u> of the <u>Apple</u> nutrients – all the antioxidants in the apple – have lost all the time when we buy it.
Ours	<u>Ninety percent</u> of the nutrition in the apple – all the antioxidant in the apple – lost all the time we bought it.
Src	随便啦……说真的， <u>阿祖力</u> 在这里 作为一个鲜活的例子，说明这个世界的故事是瞬息万变的。
Ref	Anyway – But really, <u>Azuri</u> is here as a living reminder that the story of our world is a dynamic one.
H _{TR}	Anyway, really, <u>Ayati</u> is here as a living example, and the story of this world is changing very rapidly.
H _{ET}	Any one – seriously, <u>Aris</u> is here as a living example, showing that the world is changing very rapidly.
Ours	Any one – seriously, <u>Armalie</u> is here as a living example, showing that the world is changing rapidly.

Green and red underlines indicate the correct and wrong translation of entities. The entities in references are marked by black underline

Table 8 Results of ablation study on De → En

System	Entity F_1	BLEU
Ours	63.34	35.4
– Weighted loss	62.57	35.3
– Source NER	62.27	35.0
– Target NER	61.43	35.0
– Source NER	62.77	35.4

implementation of NMT models. By enhancing the named entities, our method can ease the following problems:

- **Entity under-translation** In the second example, the location “Galapagos” appears in H_{ET} and our output, but it is missing in H_{TR} .
- **Entity over-translation** In the third example, the H_{TR} contains an extra “decades”. And in the fourth example, it generates an entity “Apple” for Apple company. However, the source sentence doesn’t have such meaning.

Table 9 BLEU scores for different α and β

α	β		
	0.3	0.6	1.0
0.3	35.2	35.4	35.1
0.6	35.3	35.2	35.1
1.0	35.1	35.2	35.2

Table 10 Entity F_1 for different α and β

α	β		
	0.3	0.6	1.0
0.3	63.53	62.61	62.46
0.6	62.78	63.34	63.16
1.0	63.91	63.05	62.61

Table 11 BLEU scores and entity F_1 for different γ value on De \rightarrow En

γ	0.5	1.0	1.5	2.0	5.0
BLEU	35.2	35.4	35.2	35.2	35.1
Entity F_1	62.29	63.34	62.78	62.95	62.61

- **Entity error-translation** In the first example, the correct name is “M. Scott Peck”. However, the baseline systems translate them as “Scott Papuk”, “MM Papker”. For numbers in the fourth example, the correct translation is “Ninety percent”, but H_{ET} result is “Ninety-nine percent”. For the fifth example, none of the system could get the name “Azuri” correct.

From these examples, we can see that all NMT system still have difficulty guaranteeing every entity is correctly translated. Fully solve this problem is still challenging and there are many potentials for this research topic.

5.5 Ablation study

To study the importance of the different parts of our system, we conduct the ablation study on De \rightarrow En translation task and the results are shown in Table 8. The minus symbol “-” means we remove the corresponding component from the system and the indentation level means the removal order. As the numbers show, when the components are gradually removed, the BLEU score and entity F_1 score will become worse. This indicates that all parts added to the system are necessary for achieving high translation performance.

Furthermore, we studied the affect of the weights of encoder and decoder NER loss, which are controlled by the hyper-parameter α and β respectively. Meanwhile, we analyzed how the hyper-parameter γ affect the translation quality. The experiments are based on IWSLT’14 De \rightarrow En dataset and the results are summarized in Tables 9, 10, and 11.

As can be seen from Tables 9 and 10, models trained with different combinations of α and β had different BLEU and entity F_1 scores. However the overall variance is small,

Table 12 Results on encoder NER ability

System	En → X	X → En
	TACC / ETACC / F_1	TACC / ETACC / F_1
En ↔ De	98.43 / 81.80 / 77.66	97.83 / 75.16 / 68.49
En ↔ Zh	98.25 / 84.66 / 77.12	98.36 / 85.45 / 77.48
En ↔ Ja	96.89 / 87.84 / 81.90	96.93 / 89.34 / 79.40

Table 13 Top and bottom three entity types in terms of F_1 on different language pairs

Rank	De → En		En → De	
	Type	F_1	Type	F_1
1	ORDINAL	76.67	GPE	76.98
2	GPE	75.60	LANGUAGE	62.86
3	LANGUAGE	70.59	CARDINAL	61.60
-3	ORG	41.03	LOC	34.15
-2	MONEY	34.69	MONEY	30.99
-1	QUANTITY	31.37	QUANTITY	28.57
Rank	Zh → En		En → Zh	
	Type	F_1	Type	F_1
1	ORDINAL	78.00	PERCENT	67.93
2	PERCENT	76.67	PRODUCT	66.67
3	GPE	59.61	GPE	65.37
-3	PERSON	21.70	PERSON	14.47
-2	EVENT	9.52	MONEY	11.11
-1	WORK_OF_ART	6.67	EVENT	3.45

which shows that our method is not sensitive to α and β . From Table 11, we can see that the performance was affected when the γ was too large or too small. Setting $\gamma = 1$ could be a good choice for this task.

5.6 Encoder entity recognition ability

We measure the encoder NER ability because high quality entity translation relies on accurate entity information extracted by the encoder. To achieve this, we extract the encoder NER output on our test set, and compare it with the ground truth extracted by DeepPavlov. Table 12 includes the accuracy of all tokens (TACC), the entity tokens only accuracy (ETACC) where all ‘0’ tags are ignored as the labels are imbalance, and the entity F_1 score. The **X** means other languages which are translated from/to English. As it shows, the encoders of all models have plausible NER ability on the inputs, with up to 98.36 on TACC and 89.34 on ETACC. Therefore, we can remove NER tools in during inference since our encoder can detect entity tokens and entity types of those tokens from the source sentences.

Table 14 Different decoding loss on De \leftrightarrow En translation

Decoding Loss	NMT		NMT + NE	
	BLEU	Entity F_1	BLEU	Entity F_1
En \rightarrow De	29.4	53.08	29.3	52.83
De \rightarrow En	35.4	63.34	35.4	63.24

5.7 Entity accuracy for different types

To study the translation quality of different entity types, we collect data from En \leftrightarrow De and En \leftrightarrow Zh test sets and sort them by F_1 score. The top and bottom three types of each language pair are shown in Table 13. Here positive rank means the better translated types and negative means the worse translated types.

The geopolitical entities (GPE), e.g. country or city names, are well translated in all language pairs. This may suggest that this type of entity is easier to learn. The “LANGUAGE” type entities in En \leftrightarrow De and “PERCENT” type entities in En \leftrightarrow Zh are also performing well. However, the “PERSON” and “EVENT” type entities are not well handled in En \leftrightarrow Zh. We suspect that is caused by the diversity of human names and the large linguistic difference between English and Chinese. We have some cases about name in Sect. 5.4 and left the way to improve it more for future study.

5.8 Test with other NER tool

Moreover, we measure the entity translation quality with Stanford NER Tagger (Finkel et al. 2005), which can detect three entity types for English: PERSON, ORGANIZATION, and LOCATION.⁵ Although the target entity types and detection algorithms are not same as DeepPavlov NER, we still have one point improvement on entity F_1 score (from 27.00 to 28.06) on De \rightarrow En dataset over standard Transformer. This implies that our methods can enhance entity translation performance under the evaluation of different NER tools.

5.9 The gap between training and decoding loss

Our method benefits from the entity loss during training. And the loss is removed in decoding time. Therefore, it is a nature question that whether such a gap will hurt the decoding performance. Especially we are using the beam search and different loss function will leads to different ranks of hypothesis. We conduct experiments on En \leftrightarrow De dataset with two decoding strategy, including translation loss only (denoted as **NMT**), and translation loss plus entity loss (denoted as **NMT + NE**). The results are shown in Table 14.

It can be witnessed that only decoding with NMT loss yields similar results as using both. Consequently, we simply decoding with NMT loss in all the experiments for efficiency.

⁵ <https://nlp.stanford.edu/software/CRF-NER.shtml>

6 Conclusions and future work

In this work, we propose a novel system to improve the translation quality of named entities for NMT, which is important for human evaluation but not well handled in previous works. The experiment results on four languages and six translation tasks demonstrate that by enhancing the encoder and the decoder with the NER ability, as well as the entity weighed loss, we can improve both entity F_1 score and BLEU score. In addition to the quality improvement, our end-to-end inference algorithm keeps the one pass decoding with little extra inference cost. This is the key difference with previous works, which rely on the NER models for translation. Therefore, it allows us to use high quality and heavy NER models but is still cost free for real world usage.

For future, there are many important possibilities that are related to this work. First, we will explore how to solve the entity translation disambiguation issue that is important for improving the translation quality. Second, we plan to study how to import external entity information, e.g. a multilingual knowledge graph to further improve entity translation. Finally, more formal theoretical analyses about using entity information in machine translation is an important direction.

Appendix A Entity types supported by DeepPavlov

Table 15 Entity types supported by DeepPavlov NER

ORGANIZATION	EVENT	PRODUCT	FACILITY	PERCENT	WORK_OF_ART
ORDINAL	LOCATION	LANGUAGE	LAW	PERSON	TIME
CARDINAL	GPE	QUANTITY	DATE	NORP	MONEY

See Table 15

The DeepPavlov support 18 different types of entity, and one special type ‘O’ to indicate non-entity tokens. All supported entity types are list in in Table 15. The details of annotation rules can be found in Weischedel et al. (2013).

Appendix B Data and processing scripts

The data and processing scripts URLs are available as follows:

- En ↔ De: <https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iwslt14.sh>
- En ↔ Zh: https://github.com/teslacoool/preprocess_iwslt/blob/master/preprocess.sh
- En ↔ Ja: <https://github.com/pmichel31415/mtnt>

Table 16 The entity F_1 and human score for En \rightarrow De

System	Entity F_1	Human (Ave.z)
Facebook_FAIR.6862	73.27	0.347
Microsoft-WMT19-sentence_document.6974	74.49	0.311
Microsoft-WMT19-document-level.6808	74.14	0.296
MSRA.MADL.6926	75.03	0.214
UCAM.6731	73.57	0.213
NEU.6763	74.34	0.208
MLLP-UPV.6651	72.74	0.189
eTranslation.6823	72.59	0.13
dfki-nmt.6479	70.38	0.119
Microsoft-WMT19-sentence-level.6785	74.00	0.094
online-B.0	72.68	0.094
JHU.6819	72.71	0.081
Helsinki-NLP.6820	72.74	0.077
online-Y.0	73.69	0.038
Imu-ctx-tf-single-en-de.6981	72.40	0.01
online-A.0	71.83	0.01
PROMT_NMT_EN-DE.6674	68.99	0.001
online-G.0	68.57	- 0.119
UdS-DFKI.6871	67.95	- 0.129
TartuNLP-c.6508	69.36	- 0.132
online-X.0	55.11	- 0.4
en_de_task.6790	37.09	- 1.769

Table 17 The entity F_1 and human score for Zh \rightarrow En

System	Entity F_1	Human (Ave.z)
Baidu-system.6940	57.79	0.295
KSAI-system.6927	58.13	0.266
MSRA.MASS.6996	59.70	0.203
NEU.6832	54.41	0.193
MSRA.MASS.6942	58.93	0.195
online-B.0	60.95	0.161
BTRANS.6825	53.55	0.186
BTRANS-ensemble.6992	54.13	0.103
online-Y.0	53.03	0.049
UEDIN.6530	49.22	0.054
NICT.6814	49.34	0.001
online-A.0	48.76	- 0.065
online-G.0	50.99	- 0.202
online-X.0	37.98	- 0.483
Apprentice-c.6706	38.02	- 0.957

Table 18 The entity F_1 and human score for En \rightarrow Cz

System	Entity F_1	Human (Ave.z)
CUNI-DocTransformer-T2T.6751	62.90	0.4020
CUNI-Transformer-T2T-2018.6457	62.95	0.4010
CUNI-Transformer-T2T-2019.6851	62.22	0.3880
CUNI-DocTransformer-Marian.6922	59.51	0.2230
uedin.6667	61.96	0.206
online-Y.0	56.80	- 0.1560
TartuNLP-c.6633	55.63	- 0.1950
online-G.0	51.57	- 0.3000
online-B.0	5.503	- 0.3360
online-A.0	46.73	- 0.5940
online-X.0	32.08	- 0.6510

Table 19 The entity F_1 and human score for En \rightarrow Ru

System	Entity F_1	Human (Ave.z)
Facebook_FAIR.6724	57.90	0.506
USTC-MCC.6795	54.05	0.332
online-G.0	52.36	0.279
online-B.0	54.35	0.269
NEU.6773	54.91	0.223
PROMT_NMT_EN-RU.6989	51.42	0.219
online-Y.0	52.12	0.156
rerank-er.6572	48.04	- 0.188
online-A.0	40.93	- 0.268
TartuNLP-u.6645	42.70	- 0.31
online-X.0	34.65	- 0.363
NICT.6563	25.77	- 1.27

Table 20 The entity F_1 and human score for Lt \rightarrow En

System	Entity F_1	Human (Ave.z)
GTCom-Primary.6998	65.43	0.234
tilde-c-nmt.6876	56.38	0.216
NEU.6759	61.99	0.213
MSRA.MASS.6945	64.03	0.206
tilde-nc-nmt.6881	57.27	0.202
online-B.0	58.44	0.107
online-A.0	49.49	- 0.056
TartuNLP-c.6908	49.04	- 0.059
online-G.0	47.02	- 0.284
JUMT.6616	39.24	- 0.377
online-X.0	34.86	- 0.396

Appendix C Entity F_1 score and human evaluation score

Tables 16, 17, 18, 19 and 20 represent the details of correlation between human evaluation standardized z score (Ave. z) and entity F_1 for English to German, Chinese to English, English to Czech, English to Russian, and Lithuanian to English, respectively.

Appendix D Human evaluation details

See Figs. 3 and 4

Fig. 3 Human evaluation score distribution

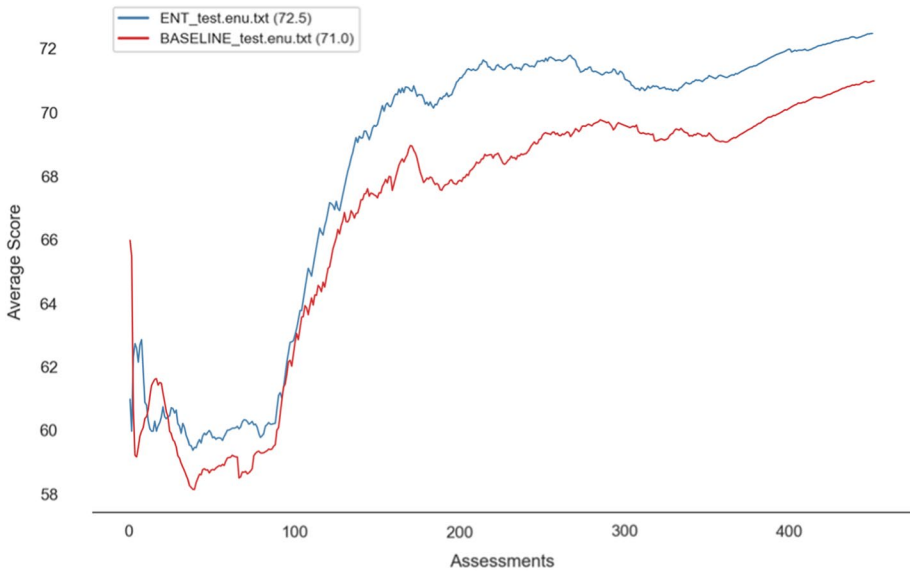
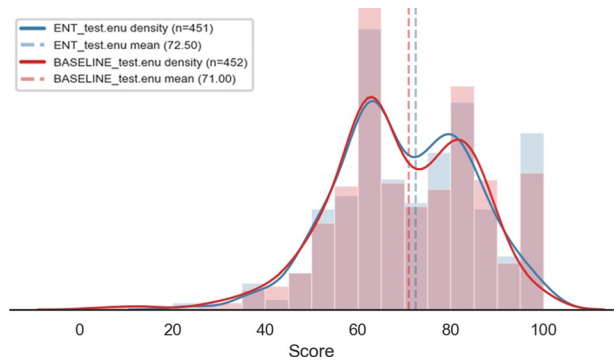


Fig. 4 Human evaluation score convergence

We used source-based direct assessment (DA) to assess translation quality. Our source-based DA method is implemented using an 88 : 12 split between data points and redundant quality controls for degraded output testing. The evaluation campaign consists of $t = 10$ tasks with $r = 1$ redundancy for $a = 5$ annotators who work on $tpa = 2$ tasks each. The score distribution is in Fig. 3 and the score convergence is in Fig. 4.

Author Contributions S. Xie, Y. Xia, L. Wu, and T. Qin conceived the idea and planned the experiments. S. Xie, Y. Huang, and Y. Fan carried out the experiments. S. Xie wrote the manuscript with support from all other authors. T. Qin supervised the project. All authors provided critical feedback and helped shape the research, analysis, and manuscript.

Availability of data and material All data sets are publicly available. The details of the URLs are in Appendix B.

Code availability Our code is available at https://www.dropbox.com/s/1owvyh6w0ahu8k4/entity_nmt.zip?dl=0

Declaration

Conflict of interest The authors declare that they have no Conflicts of interest.

References


- Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., & Turchi, M. (2017). Findings of the 2017 conference on machine translation (WMT17). In: Proceedings of the Second Conference on Machine Translation, pp. 169–214. Copenhagen, Denmark.
- Burtsev, M., Seliverstov, A., Airapetyan, R., Arkhipov, M., Baymurzina, D., Bushkov, N., Gureenkova, O., Khakhulin, T., Kuratov, Y., Kuznetsov, D., Litinsky, A., Logacheva, V., Lymar, A., Malykh, V., Petrov, M., Polulyakh, V., Pugachev, L., Sorokin, A., Vikhrev, M., & Zaynutdinov, M. (2018). DeepPavlov: Open-source library for dialogue systems. In: Proceedings of ACL 2018, System Demonstrations, pp. 122–127. Melbourne, Australia.
- Cettolo, M., Girardi, C., & Federico, M. (2012). WIT3: Web inventory of transcribed and translated talks. In: Proceedings of the 16th Annual conference of the European Association for Machine Translation, pp. 261–268. Trento, Italy
- Clinchant, S., Jung, K.W., & Nikoulina, V. (2019). On the use of bert for neural machine translation. arXiv preprint [arXiv:1909.12744](https://arxiv.org/abs/1909.12744).
- Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (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, Volume 1 (Long and Short Papers), pp. 4171–4186. Minneapolis, Minnesota.
- Dyer, C., Chahuneau, V., & Smith, N.A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 644–648. Atlanta, Georgia.
- Finkel, J.R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 363–370. Ann Arbor, Michigan.
- Fonollosa, J.A.R., Casas, N., & Costa-jussà, M. (2019). Joint source-target self attention with locality constraints. ArXiv preprint [arXiv:1905.06596](https://arxiv.org/abs/1905.06596)
- Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., & Li, M., et al. (2018). Achieving human parity on automatic chinese to english news translation. arXiv preprint [arXiv:1803.05567](https://arxiv.org/abs/1803.05567)
- Kim, Y.J., Junczys-Dowmunt, M., Hassan, H., Fikri Aji, A., Heafield, K., Grundkiewicz, R., & Bogoychev, N. (2019). From research to production and back: Ludicrously fast neural machine translation. In: Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 280–288. Hong Kong.
- Kingma, D.P., & Ba, J. (2015). Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015.

- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 Conference on EMNLP, pp. 388–395. Barcelona, Spain.
- Li, X., Yan, J., Zhang, J., & Zong, C. (2018). Neural name translation improves neural machine translation. In: China Workshop on Machine Translation, pp. 93–100. Springer.
- Li, Z., Wang, X., Aw, A.T., Chng, E.S., & Li, H. (2018). Named-entity tagging and domain adaptation for better customized translation. In: Proceedings of the Seventh Named Entities Workshop, pp. 41–46.
- Lin, T., Goyal, P., Girshick, R.B., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In: IEEE International Conference on Computer Vision, pp. 2999–3007. IEEE Computer Society.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692)
- Lu, Y., Zhang, J., & Zong, C. (2018). Exploiting knowledge graph in neural machine translation. In: China Workshop on Machine Translation, pp. 27–38. Springer.
- Luo, Y., Xiao, F., & Zhao, H. (2020). Hierarchical contextualized representation for named entity recognition. *The Thirty-Fourth AAAI Conference on Artificial Intelligence* (pp. 8441–8448). AAAI Press.
- Luong, T., Cho, K., & Manning, C.D. (2016). Neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts. Berlin, Germany.
- Michel, P., & Neubig, G. (2018). MTNT: A testbed for machine translation of noisy text. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 543–553.
- Modrzewski, M., Exel, M., Buschbeck, B., Ha, T.L., & Waibel, A. (2020). Incorporating external annotation to improve named entity translation in NMT. In: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, pp. 45–51. Lisboa, Portugal.
- Neubig, G. (2011). The kyoto free translation task
- Niehuus, J., & Cho, E. (2017). Exploiting linguistic resources for neural machine translation using multi-task learning. In: Proceedings of the Second Conference on Machine Translation, pp. 80–89.
- Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 48–53.
- Post, M. (2018). A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191. Brussels, Belgium.
- Post, M., Ding, S., Martindale, M., & Wu, W. (2019). An exploration of placeholding in neural machine translation. In: Proceedings of Machine Translation Summit XVII Volume 1: Research Track, pp. 182–192.
- Pryzant, R., Chung, Y., Jurafsky, D., & Britz, D. (2018). JESC: Japanese-English subtitle corpus. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation.
- Senrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. In: Proceedings of the First Conference on Machine Translation, pp. 83–91. Berlin, Germany.
- Senrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Berlin, Germany.
- Shavarani, H.S., Sarkar, A.: Better neural machine translation by extracting linguistic information from bert. arXiv preprint [arXiv:2104.02831](https://arxiv.org/abs/2104.02831) (2021)
- Song, K., Zhang, Y., Yu, H., Luo, W., Wang, K., & Zhang, M. (2019). Code-switching for enhancing NMT with pre-specified translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 449–459. Minneapolis, Minnesota.
- Ugawa, A., Tamura, A., Ninomiya, T., Takamura, H., & Okumura, M. (2018). Neural machine translation incorporating named entity. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3240–3250. Santa Fe, New Mexico, USA.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
- Wang, C., Cho, K., Gu, J.: Neural machine translation with byte-level subwords. arXiv preprint [arXiv:1909.03341](https://arxiv.org/abs/1909.03341) (2019)
- Wang, Y., Cheng, S., Jiang, L., Yang, J., Chen, W., Li, M., Shi, L., Wang, Y., & Yang, H. (2017). Sogou neural machine translation systems for WMT17. In: Proceedings of the Second Conference on Machine Translation, pp. 410–415. Copenhagen, Denmark.
- Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., et al. (2013). *Ontonotes release 5.0 ldc2013t19*. Philadelphia, PA: Linguistic Data Consortium.
- Wu, F., Fan, A., Baevski, A., Dauphin, Y.N., & Auli, M. (2019). Pay less attention with lightweight and dynamic convolutions. In: 7th International Conference on Learning Representations, ICLR 2019.
- Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google’s neural machine translation system: Bridging the gap between human and machine translation. [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)

- Yang, J., Wang, M., Zhou, H., Zhao, C., Zhang, W., Yu, Y., & Li, L. (2020). Towards making the most of bert in neural machine translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9378–9385.
- Zhao, Y., Xiang, L., Zhu, J., Zhang, J., Zhou, Y., & Zong, C. (2020). Knowledge graph enhanced neural machine translation via multi-task learning on sub-entity granularity. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 4495–4505.
- Zhao, Y., Zhang, J., Zhou, Y., Zong, C.: Knowledge graphs enhanced neural machine translation. In: IJCAI, pp. 4039–4045 (2020)
- Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., Liu, T.: Incorporating BERT into neural machine translation. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Shufang Xie¹  · Yingce Xia² · Lijun Wu² · Yiqing Huang³ · Yang Fan⁴ · Tao Qin²

Yingce Xia
yingce.xia@microsoft.com

Lijun Wu
lijun.wu@microsoft.com

Yiqing Huang
huang-yq17@mails.tsinghua.edu.cn

Yang Fan
fyabc@mail.ustc.edu.cn

Tao Qin
taoqin@microsoft.com

¹ Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China

² Microsoft Research, Beijing 100080, China

³ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

⁴ School of Computer Science, University of Science and Technology of China, Hefei 230026, Anhui, China