Improving Fluency Of Neural Machine Translation Using Large Language Models

Jianfei He, Wenbo Pan, Jijia Yang, Sen Peng, Xiaohua Jia
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong
{jianfeihe-2c, wenbo.pan, jijiayang2-c, senpeng.cs}@my.cityu.edu.hk
csjia@cityu.edu.hk

Abstract

Large language models (LLMs) demonstrate significant capabilities in many natural language processing tasks. However, their performance in machine translation is still behind that of the models specially trained for machine translation with an encoder-decoder architecture. This paper investigates how to improve neural machine translation (NMT) with LLMs. Our proposal is based on an empirical insight that NMT gets worse fluency than human translation. We propose to use LLMs to enhance the fluency of NMT's generation by integrating a language model at the target side. We use contrastive learning to constrain fluency so that it does not exceed the LLMs' fluency. Our experiments on three language pairs show that this method can improve the performance of NMT. Our empirical analysis further demonstrates that this method improves the fluency on the target side. Our experiments also show that some straightforward post-processing methods using LLMs, such as re-ranking and refinement, are not effective.

1 Introduction

Large Language Models (LLMs) such as GPT (Ouyang et al., 2022; Achiam et al., 2023) and LLama (Touvron et al., 2023; Dubey et al., 2024) have demonstrated significant capabilities in various domains, including language understanding and generation tasks (Chang et al., 2024). However, the evaluations (Hendy et al., 2023; Zhu et al., 2024) show that LLMs' performance in machine translation is still behind the models dedicated to the task. These dedicated models often use an encoder-decoder architecture and are trained with parallel corpora. This raises a question: Can LLMs still help improve neural machine translation (NMT)?

A key translation challenge is the balance between *adequacy* and *fluency*. According to Läubli et al. (2018), NMT is good at adequacy and weak at fluency compared to human translation. There are some *post-processing* methods to use LLMs on NMT's outputs to improve fluency. We can follow the *reranking* methods in NMT (Lee et al., 2021; Bhattacharyya et al., 2021; Fernandes et al., 2022). LLMs can be used to rerank the candidates that are output from NMT, and the one with the smallest perplexity, according to LLM's evaluation, is chosen as the final output. Alternatively, we apply the *self-refine* method in LLM (Pan et al., 2023; Li et al., 2024; Han et al., 2024) to NMT's outputs. The translations from NMT are included in the prompt and an LLM is explicitly asked to refine their fluency. These two methods are used as baselines in our experiments. Results show that they cannot consistently improve the performance of NMT.

We propose to improve the fluency of NMT's translation by integrating the language capability of LLMs during training the NMT model. We use a two-pass strategy in the decoder. The first pass is a normal one using parallel sentences. The second pass only uses the target sentences in the training data. The objective is to train a target language model while training the translation model. This is realized by assigning *all ones* to the context vectors from the encoder for the second pass. Furthermore, we use an LLM to infer the training set and get their negative log-likelihoods. These data are used with *contrastive learning* to constraint the fluency of the target language model not to exceed the LLM's.

We conduct experiments on three language pairs: German-English (De–En), Russian-English (Ru–En), and French-English (Fr–En). The results show that our method effectively improves the performance of NMT. Our empirical analysis further demonstrates that our method improves fluency on the target side, and contrastive learning with knowledge from the LLM plays an important role in achieving gains.

2 Related Work

2.1 LLMs for Translation

There is a line of research to use prompt engineering and few shot learning for LLM to translate (Zhang et al., 2023a; Gao et al., 2023). Evaluations (Hendy et al., 2023; Zhu et al., 2024) show that LLMs' performance in machine translation is still behind the NMT models dedicated to this task.

Zhang et al. (2023b), Alves et al. (2024) and Xu et al. (2024) also explore finetuning LLMs with parallel corpora to get better performance. Since LLMs have a much larger number of parameters than typical NMT, finetuning these models with a dedicated parallel corpus is not a convincing method. Such a method also does not follow the paradigm of LLMs, which aims to be general for many tasks instead of one specific downstream task.

Reranking is well investigated in the context of NMT (Lee et al., 2021; Bhattacharyya et al., 2021; Fernandes et al., 2022). The reranker is either a reference-free evaluation method such as COMET (Fernandes et al., 2022) or a dedicated trained score model in Lee et al. (2021). To the best of our knowledge, there is no research using LLMs to reranking NMT. We implement this method as one baseline in our experiments.

Using LLM to refine its own output has been investigated and is effective for some NLP tasks other than translation (Pan et al., 2023; Li et al., 2024; Han et al., 2024). Bogoychev and Chen (2023) use LLM to refine NMT's results. Their research focuses on a specific use case: terminology-aware translation.

2.2 Contrastive Learning (CL) in NLP

Contrastive Learning is applied to NMT by Yang et al. (2019) and Pan et al. (2021). However, they address specific issues. Yang et al. (2019) aim to reduce the word omission errors and Pan et al. (2021) use CL to improve the many-to-many multilingual NMT. We aim to improve the fluency of NMT, which is a more general objective.

Besides NMT, CL has applications in other NLP tasks. Sun and Li (2021) and Liu et al. (2022) apply CL for text summarization. Sun and Li (2021) use a pair-wise preference. The gold references are positive samples, and low-quality predictions are negative ones. Liu et al. (2022) use a list-wise preference. A group of ranked predictions are used in CL. These two methods work at the sequence level, while ours works at the token level.

Su et al. (2022) aim to mitigate the anisotropic distribution of token representations. They use CL to calibrate the representation space for tokens in the model.

3 Methodology

3.1 Adequacy and Fluency

Our proposal is based on the insight that NMT gets worse fluency than human translation.

There are two goals for machine translation: fluency and adequacy (Läubli et al., 2018; Kong et al., 2019; Miao et al., 2021; Sulem et al., 2020). Fluency measures whether a translation is fluent in terms of the target language. Adequacy measures whether the translation conveys the correct meaning in the source sentence, even if the translation is not fully fluent viewing from the target language.

While adequacy often requires human evaluation, fluency can be easily evaluated using the *perplexity* (denoted as *ppl*) with a language model at the target side. The relationship between perplexity and NLL (Jurafsky and Martin, 2020) is :

$$\begin{aligned} NLL &= -\sum_{i=1}^{n} log\ p(y_i|y_{<i}), \\ ppl &= e^{NLL} \end{aligned} \tag{1}$$

where y_i is the i^{th} target token and n is the total length of the target sentence.

According to Läubli et al. (2018), NMT is good at adequacy and weak at fluency compared to human evaluation. Their main result is illustrated in Figure 1.

3.2 Two-Pass Decoder

We use a two-pass procedure in the decoder in training. Each pass is related to a component in the loss function.

The first pass is through a standard decoder and gets the usual loss value of maximum likelihood estimation (MLE), which is the negative log-likelihood (NLL) with label smoothing (Edunov et al., 2018):

$$\begin{aligned} \mathcal{L}_{MLE} = &-\sum_{i=1}^{n} log\ p(y_i|X, y_{<i}) \\ &-D_{KL}(f \parallel p(y_i|X, y_{<i})), \end{aligned} \tag{2}$$

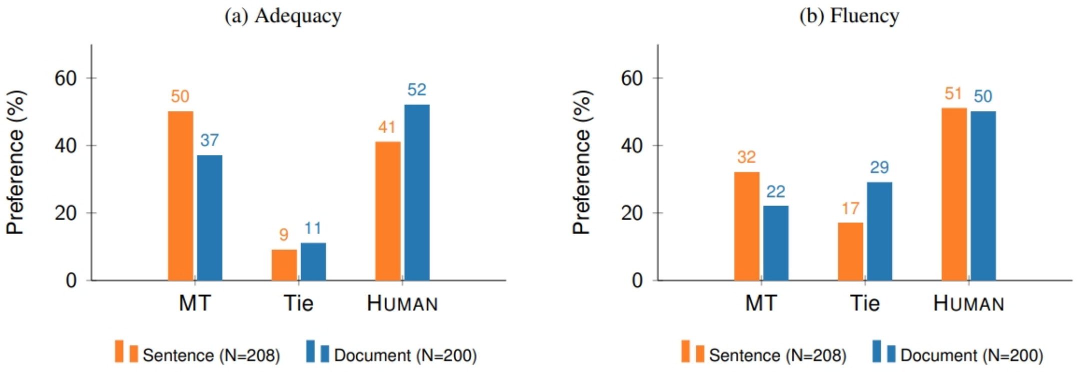


Figure 1: There is no statistically significant difference between HUMAN (human translation) and MT in terms of adequacy when evaluating sentences. However, raters show a significant preference for HUMAN in terms of fluency. From Läubli et al. (2018)

.

where X and y_i denote the source sentence and the ground truth token for step i, respectively, and f is the uniform distribution over the vocabulary. When the size of the vocabulary is V, $f = \frac{1}{V}$.

The objective of the second pass is to train the decoder to learn a target language model by *turning off* the context attention. It is realized by assigning *all ones* to the values of context vectors from the encoder. In this way, the cross-attention reduces to the query from the decoder side:

$$
\begin{aligned}
Attention(\mathcal{Q}, \mathcal{K}, \mathcal{V}) &= softmax(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \\
&= softmax(\frac{\mathcal{Q}}{\sqrt{d_k}}),\ when\ \mathcal{K}, \mathcal{V}\ are\ all\ ones.
\end{aligned} \tag{3}
$$

Correspondingly, this second pass gets the second loss component:

$$
\mathcal{L}_{fluency} = -\sum_{i=1}^{n} log\ p(y_i|y_{<i}) \tag{4}
$$

In this two-pass procedure, the same network architecture is used, and all parameters are shared.

This is a potential conflict between the $\mathcal{L}_{fluency}$ in Equation 4 and $\mathcal{L}_{MLE}$ in Equation 2. When the model is trained using the loss component in Equation 4, $log\ p(y_i|y_{<i})$ is maximized. This may conflict with the translation objective in Equation 2 which maximizes $log\ p(y_i|y_{<i}, X)$. We use *contrastive learning* to mitigate this conflict.

3.3 Contrastive Fluency Enhancement (CFE)

Contrastive Learning (CL) has a key component: a *max* function. It is defined as:

$$
\max\{0, \rho + S_n - S_p\}, \tag{5}
$$

where S_n and S_p are scores for negative and positive samples, respectively. ρ is a hyperparameter, the margin between the scores between negative and positive samples.

This function outputs a positive loss when the score of the negative sample is larger than one margin plus the score of the positive sample. The objective is to constrain the score of the negative sample so that it is at least one margin lower than the score of the positive sample.

We use the negative log-likelihood (NLL) of target tokens as the scores. The values from the training models are negative samples, while those from LLMs are positive samples. This method is denoted as Contrastive Fluency Enhancement (CFE) and the corresponding loss component is:

$$
\begin{aligned}
&\mathcal{L}_{CFE} = \\
&\max\{0, \rho - \sum_{i=1}^{n} log\ p(y_i|y_{<i}) + \sum_{i=1}^{n} log\ p_{llm}(y_i|y_{<i})\},
\end{aligned} \tag{6}
$$

where p_{llm} is the probability in LLM.

The final loss function is:

$$
\mathcal{L} = \mathcal{L}_{MLE} + \mathcal{L}_{CFE} \tag{7}
$$

To conduct the ablation study, we implemented a variant without CL. Its loss function is:

$$
\mathcal{L}_{MLE} + \mathcal{L}_{fluency} = \mathcal{L}_{MLE} - \sum_{i=1}^{n} log\ p(y_i|y_{<i}) \tag{8}
$$

4 Experiments

4.1 Datasets

We use the negative log-likelihood (NLL) from an LLM as the positive samples during training. The major data used to train an LLM are usually in English. Therefore, we use English as the target language in our experiments.

We use the corpora from WMT[1] as our datasets.

We use Europarl v7, News-commentary-v12, and Common Crawl for training in De–En. The training data have totally 4.6 million sentences. We use Newstest2014 for validation, and Newstest2021 for testing in De–En. For Ru–En, ParaCrawl v9, News-commentary-v10, and Common Crawl are used for training. These training data have totally 13.1 million sentences. Newstest2014 is used for validation, Newstest2021 is used for testing in Ru–En. For Fr–En, Europarl v7, News-commentary-v10, and Common Crawl are used for training. These training data have totally 5.4 million sentences. Newstest2013 is used for validation, and Newstest2015 is used for testing in Fr–En.

We need to use an LLM to infer each target sentence in the training set to get its negative log-likelihood. Therefore, we limit the size of the training set by filtering the original datasets. We randomly select 350 million sentences from the original training dataset for each language pair. We use the condition below to further choose data with high quality:

- Both source and target sentences have lengths within the range of 5 to 300.
- The disparity between the source and target sentence length does not exceed five times.

The number of sentence pairs for each language pair is as follows: De–En 2.6 million, Ru–En 2.9 million, Fr–En 2.7 million.

4.2 Systems

We compare our method with the vanilla Transformer model, three typical token-level methods improving NMT, and two methods introduced in Section 2 for comparison. Our method is not compared with sequence-level methods such as *MIXER* (Ranzato et al., 2016) and *MRT* (Shen et al., 2016). These sequence-level methods use online samples and are more than ten times slower than the token-level methods (Edunov et al., 2018).

- *TX* is the vanilla Transformer.
- *SS* (Mihaylova and Martins, 2019) is a scheduled sampling method with a Transformer that uses two-pass decoding. The Inverse Sigmoid Decay is used for scheduling in our experiments. It performs best among the scheduling algorithms according to Liu et al. (2021).
- *CASS* (Liu et al., 2021) is Confidence-Aware Scheduled Sampling. It enhances the normal scheduled sampling by sampling different tokens according to the model's probability of ground truth tokens.
- *TFN* (Goodman et al., 2020) uses two stacking decoders. The loss values are computed on each decoder and the results are combined to form the final loss value. We use the hyperparameters according to their recommendation in the paper. The second decoder's weight is set to 0.4, and both decoders share the same set of parameters.
- *Refine* includes the translations from NMT in the prompt and explicitly asks LLM to refine the fluency.
- *ReRank* uses LLMs to rerank the output candidates from NMT and choose the one with the smallest perplexity in LLM.

We implement our proposal, Contrastive Fluency Enhancement (CFE), as described in Section 3.

Since *ReRank* is a post-processing method, we can apply ReRank to the output of CFE. This variant is denoted as *CFE+ReRank*.

4.3 Implementation Details

We use Llama2-13B-chat-hf [2] as the LLM for experiments. Its negative log-likelihood of each token in the target sentences in the training data is used as described in Section3.3. For the method *Refine*, this model is also used to generate refined translations. In inference, we use top-p (0.9) sampling, and the sampling temperature is set to 0.9.

Our implementation of NMT is based on the Fairseq toolkit (Ott et al., 2019) using a typical configuration [3] similar to the original Transformer (Vaswani et al., 2017). The Transformer Base model with about 60 million parameters is used. Since we use the token-level negative log-likelihood from Llama2-13B-chat-hf, we need to

[1] http://www.statmt.org

[2] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[3] https://github.com/facebookresearch/fairseq/tree/main/examples/scaling_nmt

	De–En			Ru–En			Fr–En		
Metrics	BLEU	Meteor	Comet	BLEU	Meteor	Comet	BLEU	Meteor	Comet
Baselines									
Transformer	26.19	49.18	75.45	28.76	49.98	75.28	34.41	51.17	76.51
SS	26.43	49.20	75.40	28.71	49.82	74.99	34.55	51.12	76.29
CASS	26.27	**49.54**	75.56	28.96	50.14	75.30	35.14	51.72	76.67
TFN	26.31	**49.54**	75.44	28.99	50.23	75.30	34.32	51.13	76.67
Refine	26.19	49.18	75.45	28.76	49.98	75.28	34.41	51.17	76.51
ReRank	26.42	**49.90**	75.76	28.99	**51.05**	75.93	*33.09*	*51.08*	*76.06*
∆ (-TX)	0.23	0.72	0.31	0.23	1.07	0.65	-1.32	-0.09	-0.45
Our Proposal									
CFE	**26.65**	**49.45**	**75.91**	**29.67**	**50.82**	**76.51**	**35.50**	**51.88**	**76.86**
∆ (-TX)	0.46	0.27	0.46	0.91	0.84	1.23	1.09	0.71	0.35
CFE+ReRank	**27.06**	**50.18**	**76.03**	**29.72**	**51.73**	**76.87**	*33.76*	51.74	*76.12*
∆ (-TX)	0.87	1.00	0.58	0.96	1.75	1.59	-0.65	0.57	-0.39

Table 1: Performance of different methods. The scores of CFE and those better than CFE are highlighted in **Bold**, while the scores that are worse than the vanilla Transformer (denoted as **TX**) are shown in *Italic*. ∆ denotes the gain compared to TX.

use the same tokenizer for NMT and Llama2-13B-chat-hf so that one sentence has the same subwords in two systems. We use the tokenizer of Llama2-13B-chat-hf for subwords. The vocabulary size is equal to 32,000, which is shared for the source and target sentences. Both the dropout rate and the label smoothing are set to 0.1. We use beam search for decoding with a beam size of six, and the factor for length penalty is 0.6. The number of candidates used for *ReRank* is the same as the beam size.

In our preliminary experiments for *Refine*, we found that the outputs from LLMs may contain some explanation words. This result makes it difficult to extract the refined sentence for evaluation. Therefore, the prompt used for *Refine* in our evaluation requires that the LLM do not give any explanation. The prompt is shown below:

"initial translation"

If there are minor mistakes in the above sentence, please correct them and make this sentence more fluent. If there is no mistake, keep it intact. Only output the result. No explanation.

Our method, its variant for ablation study, and token-level baseline methods (SS, CASS, TFN) use a common pre-trained NMT model for finetuning. This pre-trained model is trained for a minimum of 20 epochs on the filtered data set described in Section 4.1, stopping if the validation loss does not decrease for 20 consecutive epochs. For finetuning, we adopt the same early-stop policy as Choshen et al. (2019), where the process is terminated if the validation loss does not decrease for ten consecutive epochs. The margin ρ in the loss function of CFE is set to 0.1.

All GPUs used for training are Nvidia GF1080Ti.

4.4 Evaluation and Results

Three metrics are used to evaluate the performance of the methods using: BLEU, Meteor, and Comet. We use SacreBLEU[4] (Post, 2018)[5] for BLEU. For Meteor[6], we use its version 1.5. For Comet, we use its *wmt22-comet-da* model[7].

Table 1 illustrates the performance of methods for De–En, Ru–En, and Fr–En.

The vanilla Transformer model is a strong baseline. Our method CFE outperforms it in all three metrics for all language pairs. CPE generally achieves the best performance compared to other baselines except for a few cases in Meteor.

Refine gets the same performance as the vanilla Transformer. We find that LLM almost always regards the translation from NMT as fluent enough and does not provide improved translations. The number of *intact* sentences are illustrated in Table 4.

ReRank gets better performance than the vanilla Transformer for De–En and Ru–En, but much

[4] https://github.com/mjpost/sacreBLEU

[5] case.mixed+numrefs.1+smooth.exp+tok.13a+version.2.3.1

[6] http://www.cs.cmu.edu/~alavie/METEOR/

[7] https://github.com/Unbabel/COMET

worse for Fr–En. Table 2 and 3 illustrate that ReRank always gets much lower perplexity than the vanilla Transformer. The inconsistency between low perplexity and good translation reflects the complexity of machine translation and the importance of the balance between adequacy and fluency.

CPE+ReRank gets gains in De–En and Ru–En. However it has worse performance than CPE in Fr–En. This result is consistent with the bad performance of ReRank alone in Fr–En.

	Model	De–En	Ru–En	Fr–En
ppl	TX	217.9	128.5	242.3
	ReRank	73.0	62.4	94.9
	CFE	117.6	131.5	223.0
	CFE+ReRank	72.1	66.1	87.8
NLL	TX	4.131	3.923	4.406
	ReRank	3.823	3.631	4.019
	CFE	4.108	3.895	4.404
	CFE+ReRank	3.798	3.602	3.999

Table 2: Fluency measured with average perplexity (ppl) and negative log-likelihood (NLL).

	De–En	Ru–En	Fr–En
Better	855	835	1301
Equal	145	165	195
Worse	0	0	0

(a) ReRank, compared to Transformer

	De–En	Ru–En	Fr–En
Better	477	486	578
Equal	95	93	346
Worse	428	421	572

(b) CFE, compared to Transformer

	De–En	Ru–En	Fr–En
Better	775	767	1174
Equal	37	38	113
Worse	188	195	209

(c) CFE+ReRank, compared to Transformer

Table 3: Investigate the fluency compared to Transformer at sentence-level using negative log-likelihood.

5 Analysis

5.1 Loss Components in CFE

Figure 2 shows the components in the loss function of CFE for De–En during training. Both the loss component $\mathcal{L}_{fluency}$(Figure 2a) and the total loss (Figure 2b) steadily decrease. These figures demonstrate the effectiveness of the CFE loss function presented in Section 3.

The results on other language pairs get to the same conclusion as illustrated in Figure 3 and 4.

5.2 Fluency

The fluency usually is measured with *perplexity*, denoted as *ppl*. We use Llama2-13B-chat-hf to get the NLL of each translation, which is averaged based on the number of tokens in the generated sentence. These NLLs are used to calculate that sentence's perplexity according to Equation 1.

Table 2 illustrates each test set's average perplexity and NLL. ReRank outputs the one with the lowest NLL in the candidates. Therefore, it consistently gets much lower perplexity and NLL compared with the vanilla Transformer, even for Fr–En that ReRank gets much worse performance as shown in Table 1.

Our method CFE consistently gets lower NLL for all language pairs than the vanilla Transformer. CFE generally gets a lower average perplexity, with the only exception being Ru–En. Compared to ReRank, CFE gets larger perplexity and NLL. This result reflects that CFE gets a better balance between fluency and adequacy.

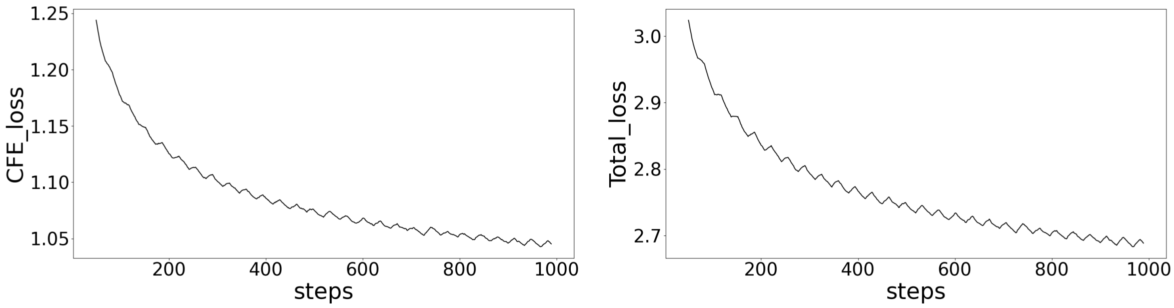
We also compare the NLL of the vanilla Transformer and other methods for each translation and count the number of cases that other methods have lower (*better*), equal, or greater (*worse*) NLL than the vanilla transformer. When the absolute value of the difference in comparison is less than 0.001, two NLL values are counted as *equal*. The results illustrated in Table 3 show that our method effectively improves the fluency of NMT.

5.3 Refine With LLM

Table 4 illustrates that most sentences are kept intact when the LLM is asked to improve fluency. There are a few sentences in which no translations are identified in the feedback from Llama2-13B-chat-hf. When these *empty* feedback are identified, the original translations are reasonably used before evaluation in our implementation. This analysis explains why *Refine* gets the exactly same performance as the vanilla Transformer.

5.4 Ablation Study

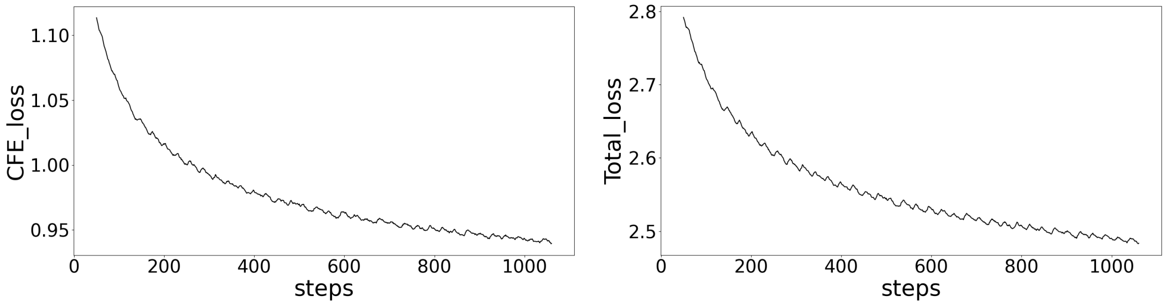
Table 5 shows the performance of CFE with and without Contrastive Learning (CL). The variant without CL implements the loss function in Equation 8. It maximizes the target language model



(a) Loss component of CFE

(b) Total loss

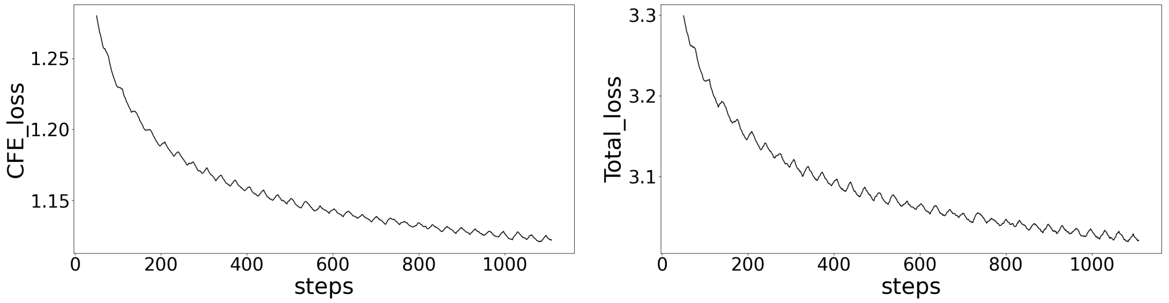
Figure 2: Investigate the components in the loss function for De–En



(a) Loss component of CFE

(b) Total loss

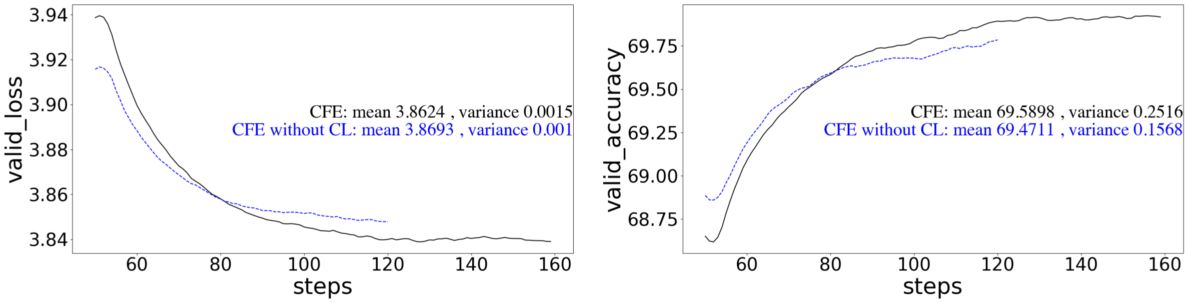
Figure 3: Investigate the components in the loss function for Ru–En



(a) Loss component of CFE

(b) Total loss

Figure 4: Investigate the components in the loss function for Fr–En

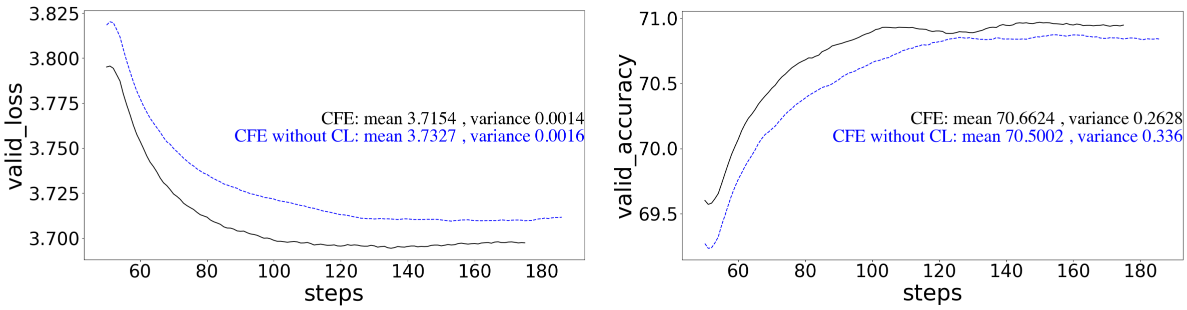


(a) Loss on validation set

(b) Accuracy on validation set

Figure 5: Investigate the performance on the validation sets during training for CFE and its variant without contrastive learning for De–En.

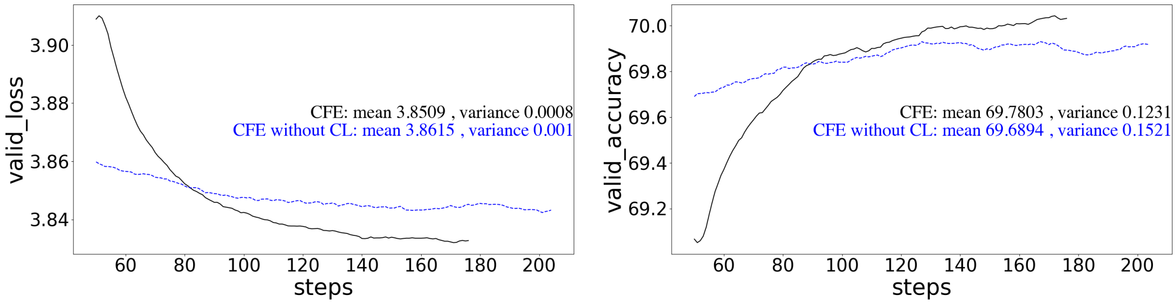
and does not make use of LLM's knowledge as a ceiling. While CFE without CL also outperforms the vanilla Transformer model and demonstrates its efficacy in improving NMT, its gains are generally



(a) Loss on validation set

(b) Accuracy on validation set

Figure 6: Investigate the performance on the validation sets during training for CFE and its variant without contrastive learning for Ru–En.



(a) Loss on validation set

(b) Accuracy on validation set

Figure 7: Investigate the performance on the validation sets during training for CFE and its variant without contrastive learning for Fr–En.

	De–En	**Ru–En**	**Fr–En**
Total	1000	1000	1496
Intact	995	996	1480
Empty	5	4	16

Table 4: *Refine* with the LLM does not improve NMT.

Model	**De–En**	**Ru–En**	**Fr–En**
TX	26.19	28.76	34.41
CFE	26.65	29.67	35.50
Δ **(-TX)**	0.46	0.91	1.09
w/o-CL	26.58	29.01	34.66
Δ **(-TX)**	0.39	0.25	0.25

Table 5: Ablation test by removing Contrastive Learning from CFE, denoted as *w/o-CL*.

lower than CFE.

Figure 5 shows the performance on the validation sets during training for CFE (black and solid) and its variant (blue and dashed) without CL in De–En. It shows that the variant consistently gets higher loss and lower accuracy during training. Figure 6 and 7 illustrate the performance on the validation set for the other language pairs, which are consistent with the conclusion of De–En.

These ablation tests demonstrate the importance of Contrastive Learning in CFE.

5.5 Significance Tests

Table 6 shows the results of significance tests for *ReRank*, *CFE* and *CFE+ReRank* (denoted as *CFE+RR+ST*). We report mean and standard error over five training runs with seeds 1–5. For *ReRank*, these seeds are applied to pretrained models. These results are generally consistent with Table 1.

Model	**BLEU**	**Meteor**	**Comet**
TX	26.19	49.18	75.45
ReRank-ST	26.37 ±.11	49.88 ±.09	75.70 ±.06
Δ **(-TX)**	0.18	0.70	0.25
CFE-ST	26.65±.09	49.37 ±.06	75.87 ±.07
Δ **(-TX)**	0.46	0.19	0.42
CFE+RR-ST	26.70±.11	49.84±.11	75.85 ±.09
Δ **(-TX)**	0.51	0.66	0.40

Table 6: Significance tests on De–En.

6 Conclusion

This paper investigates how to improve neural machine translation (NMT) with Large language models (LLMs). Our experiments show that post-processing methods like re-ranking and self-refining are not effective. Based on the insight that NMT is good at adequacy and weak at fluency, we propose to use LLMs to enhance the fluency of NMT's generation by integrating a language model at the target side and using Contrastive learning to constraint the probabilities to a ceiling, the LLM's fluency. Our experiments on three language pairs (De–En, Ru–En, and Fr–En) show that this method effectively improves the performance of NMT. The empirical analysis further demonstrates that this method improves the fluency at the target side and Contrastive Learning with knowledge from the LLM plays an important role in achieving the gains.

7 Sustainability Statement

We trained and finetuned the model with the early-stop strategy as described in Section 4.3. Pretraining and finetuning the model typically took nearly 140 and 100 GPU-hours using Nvidia GF1080Ti. The estimated energy cost for each model is illustrated in Table 7, according to the calculation using Green-Algorithms[8].

	GPU-Hour	CO_2(kg)	Engergy(kWh)
Pretrain	140	31.88	59.32
Finetune	110	25.05	46.61

Table 7: Estimated energy cost for each model.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Duarte M Alves, José Pombal, Nuno Miguel Guerreiro, Pedro Henrique Martins, João Alves, M Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *CoRR*.

Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2021. Energy-based reranking: Improving neural machine translation using energy-based models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Nikolay Bogoychev and Pinzhen Chen. 2023. Terminology-aware translation with constrained decoding and large language model prompting. In *Proceedings of the Eighth Conference on Machine Translation*, pages 890–896.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2019. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. 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 355–364.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Yuan Gao, Ruili Wang, and Feng Hou. 2023. How to design translation prompts for chatgpt: An empirical study. *Preprint*, arXiv:2304.02182.

Sebastian Goodman, Nan Ding, and Radu Soricut. 2020. Teaforn: Teacher-forcing with n-grams. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8704–8717.

Haixia Han, Jiaqing Liang, Jie Shi, Qianyu He, and Yanghua Xiao. 2024. Small language model can self-correct. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18162–18170.

[8] http://calculator.green-algorithms.org

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Daniel Jurafsky and James H Martin. 2020. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Xiang Kong, Zhaopeng Tu, Shuming Shi, Eduard Hovy, and Tong Zhang. 2019. Neural machine translation with adequacy-oriented learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6618–6625.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7250–7264, Online. Association for Computational Linguistics.

Loka Li, Guangyi Chen, Yusheng Su, Zhenhao Chen, Yixuan Zhang, Eric Xing, and Kun Zhang. 2024. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *CoRR*.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Confidence-aware scheduled sampling for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2327–2337.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903.

Mengqi Miao, Fandong Meng, Yijin Liu, Xiao-Hua Zhou, and Jie Zhou. 2021. Prevent the language model from being overconfident in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3456–3468.

Tsvetomila Mihaylova and André FT Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Shiqi Shen, Yong Cheng, Zhongjun 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 (Volume 1: Long Papers)*, pages 1683–1692.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Elior Sulem, Omri Abend, and Ari Rappoport. 2020. Semantic structural decomposition for neural machine translation. In *Proceedings of the ninth joint conference on lexical and computational semantics*, pages 50–57.

Shichao Sun and Wenjie Li. 2021. Alleviating exposure bias via contrastive learning for abstractive text summarization. *arXiv preprint arXiv:2108.11846*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023b. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with qlora. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781.