Prompting Large Language Models with Human Error Markings for Self-Correcting Machine Translation

Nathaniel Berger[a*] and Stefan Riezler[ab]
Computational Linguistics[a] & IWR[b]
Heidelberg University
69120 Heidelberg, Germany
berger@cl.uni-heidelberg.de
riezler@cl.uni-heidelberg.de

Miriam Exel[c] and Matthias Huck[c]
SAP SE[c]
Dietmar-Hopp-Allee 16
69190 Walldorf, Germany
miriam.exel@sap.com
matthias.huck@sap.com

Abstract

While large language models (LLMs) pre-trained on massive amounts of unpaired language data have reached the state-of-the-art in machine translation (MT) of general domain texts, post-editing (PE) is still required to correct errors and to enhance term translation quality in specialized domains. In this paper we present a pilot study of enhancing translation memories (TM) produced by PE (source segments, machine translations, and reference translations, henceforth called PE-TM) for the needs of correct and consistent term translation in technical domains.

We investigate a light-weight two-step scenario where, at inference time, a human translator marks errors in the first translation step, and in a second step a few similar examples are extracted from the PE-TM to prompt an LLM. Our experiment shows that the additional effort of augmenting translations with human error markings guides the LLM to focus on a correction of the marked errors, yielding consistent improvements over automatic PE (APE) and MT from scratch.

1 Introduction

Technical translation at large enterprises involves a large number of translation domains, for which translation memories and terminologies need to be maintained to support multi-domain MT systems and human post-editors in producing contextually adequate and consistent translation of technical terms (Exel et al., 2020). In this paper, we ask if ongoing human post-editing efforts that produce large databases consisting of source segments, machine translations, and reference translations, can be enhanced by light-weight human error markings. This could then be used to teach a translation system a focused self-correction of marked erroneous tokens from similar examples with error markings and corrections found in the PE-TM. Such a setup could complement translation memories and terminology databases by up-to-date and domain-specific information in the PE-TM, and be used in a scenario where a user marks errors in MT hypotheses. In-context examples with high source-side similarity are then extracted from the PE-TM to prompt an LLM to focus on a correction of the marked error interactively.

We present a pilot study where we construct a PE-TM for the IT domain, which is augmented by human error markings on machine translations. While for training purposes, error markings for the PE-TM could be obtained by automatic matching against human post-edits, this cannot be done at test time. We envisage a scenario where the error markings in the PE-TM are obtained by direct human annotation, simulating a realistic setup where a user only performs the light-weight task of error marking at test time. Such a scenario could be feedback collection in the publishing of raw-MT. Raw-MT could be shown to end-users who, if they notice an error in the translation, proceed to annotate tokens in the translation they perceive to be incorrect. The translation would then be flagged for review by a human translator, who then post-edits the translation and publishes their correction. This process results

Figure 1: Example of a 1-shot prompt for English-to-German Translation. Error markings are inside bold faced tags <**bad**> </**bad**>. The demonstration example consists of a source segment in English (in green), a translation hypothesis in German (in blue), and a correction (in red). The test example shows a correction of the translation of "environment variable" from "Umweltvariable" into "Umgebungsvariable" learned by the LLM (in bold-faced red).

in the creation of (source, hypothesis, post-edit) triples with annotations for the PE-TM. This PE-TM is used to provide in-context examples for LLM correction of annotated translation hypotheses. For example, the end-user who annotates raw-MT could then immediately be shown a new translation that takes the error markings into account.

The results of our study show that selecting in-context examples based on similarity of source-side embeddings and providing error markings on hypotheses lets the LLM infer focused corrections of marked errors. Furthermore, overall translation quality is improved over few-shot prompt-based translation and over automatic post-editing. An example 1-shot prompt and error-marked output is given in Figure 1.

2 Prior Work

The last year has seen a progression of the translation capabilities of decoder-only LLMs, pre-trained on unpaired language data, from lagging behind supervised systems (Vilar et al., 2023) to matching their translation quality (Garcia et al., 2023), with only 5 examples of high-quality translation data used for in-context learning. However, MT in specialized domains still requires translation post-editing in order to correct errors and to enhance term translation quality. Raunak et al., (2023) recently showed that very large LLMs (OpenAI,

2023) can perform zero-shot automatic translation post-editing for general domain data, at the price of hallucinated edits. This makes this setup impractical if high precision in domain-specific translation is key. For these purposes, manually crafted glossaries (Vidal et al., 2022), dictionaries extracted in a separate step of unsupervised word-alignment (Ghazvininejad et al., 2023), or translation memories accessed with fuzzy matching (Moslem et al., 2023; Hoang et al., 2023), have been used to aid prompt-based MT. Our approach combines PE-TMs with light-weight human error markings, achieving improvements over both APE and MT from scratch.

The standard paradigm to incorporate token-level human error markings as learning signal is an adaptation of supervised learning from post-edits (see, for example, Turchi et al. (2017)) by penalizing erroneous tokens and rewarding correct tokens in a weighted maximum-likelihood objective (Marie and Max, 2015; Domingo et al., 2017; Petrushkov et al., 2018; Lam et al., 2019; Kreutzer et al., 2020; Berger et al., 2023). Most approaches are conceptualized as fine-tuning applications, with error markings obtained by automatic matching against human post-edits or by direct human annotation. The approach that is closest to our work is QuickEdit (Grangier and Auli, 2018). They train a model with separate encoders for source and error-marked

hypothesis in order to improve upon the initial hypothesis by avoiding the marked tokens. Similar to our approach, QuickEdit requires error-markings at inference time. While QuickEdit relies on supervised learning, our approach succeeds in teaching an LLM to avoid marked tokens from a few demonstration examples of similar error patterns.

More recent work by Xu et al., (2023) successfully uses feedback in form of error type and location that is predicted by a learned error pinpoint model. Their work focuses on general domain translation and quality-estimation type feedback, in difference to the focused error markings on technical terms that we are interested in. We plan an extension of our work in the direction of using learned error markings in future work.

Our work is furthermore related to the more general issue of self-correction capabilities of LLMs. Similar to the findings of Huang et al., (2023), our work shows that in order to qualify as a correction rather than a mere change, automatic self-correction in LLMs (Madaan et al., 2023; Pan et al., 2023) needs to be guided by an oracle. In our case, the oracle consists of feedback on the errors in translation outputs of an LLM, combined with a few examples of similar errors and their reference translations.

3 Data and Models

We collected English and German parallel data from open source software documentation and localization available on OPUS (Tiedemann, 2012), as this data comes closest to our domain of interest. We concatenated data from *GNOME*, *KDE4*, *KDEdoc*, *PHP*, and *Ubuntu* to create our data set and filtered them with the following methods: we removed those segments containing fewer than five words or more than 25, those identified by fastText (Joulin et al., 2017) as the wrong language, those with more than 20% of characters being non-alphanumeric, and those containing personally identifiable information. Of the remaining data, we selected a subset of $1,500$ examples.

For the purposes of this experiment, we were interested in models that support prompt-based interaction. Furthermore, we are interested in the scenario where users use their judgment to guide a model towards a better translation based upon its original translation. Large language models lend themselves well to this interaction because the same model can be used with prompt-based interaction to produce the original translations as well as for pro-

viding extra information to aid in correction. These considerations decide in favor of using an LLM over a traditional encoder-decoder based model typically used in production scenarios. Therefore, we examine if the model that produced the hypothesis, Llama 13B (Touvron et al., 2023), can leverage feedback to correct its own mistakes. In addition to choosing this model because it supports prompting, it runs on a single GPU[1], and the model will remain available in the future for reproducibility. The Llama model was converted to Huggingface Transformers (Wolf et al., 2020) format for inference. We found that Llama 13B frequently copies hypotheses including the error tags to its output as it was instructed not to make changes if the hypothesis is acceptable as-is. We therefore post-processed outputs by removing tags by regex.

Additionally, we test GPT-3.5[2] in order to test a model larger than we can locally run. We use the "ChatCompletion" API, send the query as the 'user' message, and set the temperature to 0. All hypotheses were generated with the above models using greedy decoding.

4 Feedback Collection

In order to simulate the previously proposed scenario of an end-user who annotates raw-MT errors, we turn to paid annotators. We generate translations of English source sentences and provide them with only the source and the hypothesis, as would be the case when getting feedback on raw-MT. These are then paired with references to create our PE-TM.

4.1 Human Annotation

We hired three professional translators with expertise in the IT domain as annotators to provide token-level feedback on the translation hypotheses. Token-level feedback consisted of per-token binary quality judgements, OK/BAD. Annotators were provided English source sentences and German hypothesis translations in a custom annotation interface. Each token in the hypothesis was a button in the annotation interface and annotators were instructed to click on incorrect tokens to mark errors. Unmarked tokens were assumed OK. Additionally, they were instructed to keep markings minimal and only mark tokens that would be edited or deleted during post-editing.

---

[1]For all Llama 13B experiments, we use a single Nvidia A40 GPU with 48GB VRAM on a shared server
[2]GPT-3.5-Turbo-0613 was used for all experiments involving OpenAI's GPT models in this paper

You will receive an annotation task called "Error Annotation"; the goal of this task is to mark each word in the machine translation as correct or incorrect. By default, all words are considered correct. By clicking on target words, they are marked as incorrect.

Here are the instructions in more detail:
You will be shown an English source sentence above and its machine translation into German below.

- Begin by reading the source sentence and then reading the translation.

- Consider which words would need to be deleted or changed in order to arrive at a correct translation.

- Mark the incorrect words of the translation by clicking on them.

- Clicking on the word causes a blue border to appear around the word. This word is now marked as incorrect.

- Clicking a second time will remove the blue border and it is now marked correct.

- Once all the incorrect words have a blue border, click on the "Next" button near the top of the page.

- Markings should be kept minimal. Mark only those terms that you would edit or delete in a post-editing scenario.

- If you would have to move a word to a different location, such as shifting a verb to the end of the sentence, mark it as incorrect.

If the translation contains no correct words or the source words are translated word by word but do not make sense together, mark them all as incorrect.

If the translation is correct as-is, proceed to the next annotation item.

If you cannot judge the quality of the translation because the source sentence is not comprehensible, or you are lacking domain knowledge to annotate wrong words, click the Skip button (to the right of the "Next" button) and then proceed to the next sentence.

The source sentences are taken from open-source software projects and documentation while the translations are produced by a generic machine translation system.

Figure 2: Instructions given to annotators on how to mark errors in sentences, including how to use the interface and desired marking behavior

Complete instructions are shown in Figure 2. Annotators could skip examples but must provide a reason. Reasons for skipping examples were "Source Incomprehensible", "Source Ambiguous", "Missing Knowledge", and "Other".

Annotation was split into two phases. Phase one was a trial run where all three annotators annotated the same 50 examples. In phase two, each annotator was given their own non-overlapping block of 500 source and hypothesis pairs. The phase one examples were used to compute summary statistics of annotation behavior, agreement coefficients, and to calibrate our instructions.

After phase two, filtering out skipped examples or those without any BAD markings yields a data set of 982 examples. We split this data set into two subsets; one set of size 492 for in-context examples and a set of 490 for test examples.

4.2 Annotation Statistics

Annotator 1 selected "Source Ambiguous" as the reason for skipping once and "Missing Knowledge" the other six times. Annotator 2 selected "Source Incomprehensible" for their skip. After removing the items skipped by any annotator, we have 43 examples that were annotated by all three.

Using the remaining common examples from phase one, we calculate the percentage of tokens marked per sentence and use that as a sentence-level quality judgment. This is then used to calculate Krippendorff's alpha (Krippendorff, 2004) to determine if our annotators agree on overall translation quality. We also calculate alpha on the token level OK/BAD annotations.

We calculated pair-wise Krippendorff's alpha in addition to the average agreement for both the sentence-level percentage marked and token-level annotations. The average amount of tokens marked for the unskipped sentences is visible in Table 1. Pairwise Krippendorff's alphas for percentage

Annotator	1	2	3
Percent Marked on Average	0.25	0.17	0.17
SD of Percent Marked	0.28	0.18	0.19

Table 1: Marking behaviors of each annotator in terms of percent of tokens marked in the trial annotation.

Annotator	2	3
1	0.258	0.481
2	∅	0.222

Table 2: Inter-annotator agreement for percentage marked per sentence, given by Krippendorff's Alpha.

Annotator	2	3
1	0.445	0.531
2	∅	0.433

Table 3: Inter-annotator agreement for token classification, given by Krippendorff's Alpha.

Annotator	1	2	3
Percent Marked on Average	0.10	0.19	0.09
SD of Percent Marked	0.10	0.15	0.1

Table 4: Marking behaviors of each annotator in terms of percent of tokens marked in the final annotation.

marked is visible in Table 2, while pairwise agreement for token classification is in Table 3. Average agreement for the percentage marked is $\alpha = 0.306$ and for token classification $\alpha = 0.466$. This suggests that, while agreement about overall sentence quality is not high, the reliability of classifying each token in the hypothesis is higher. These results were used to calibrate with the annotators after looking over the annotations made by each individual.

After calibration, we then assigned each annotator their block of 500 examples to annotate. Annotator 1 skipped 6 of the 500 sentences and annotator 2 skipped 20. Percentage marked was lower for annotators 1 and 3 during the full annotation as more sentences were left completely unmarked. Annotator 1 left 36% of sentences unmarked; annotator 2 left 23%; and annotator 3 left 38%. The percentage that was marked per sentence was also reduced after calibration, as shown in Table 4.

5 Experiments

5.1 Experimental Setup

Using the annotated data, we considered three machine translation tasks: Machine translation from scratch (*MT*); Automatic Post-editing (*APE*); and Post-Editing with error markings (*MRK*). Instructions were written for the LLM for each task and, for each example in the inference set, five examples were retrieved from the in-context example pool. We retrieve the most similar examples by using cosine similarity over SentenceTransformers (Reimers and Gurevych, 2019) embeddings computed on source sentences only[3].

In *MT*, models were prompted to

Translate English to German.

and were shown five (source, reference) pairs. Full prompts can be found in the appendix A.1. For the *APE* task, models were prompted to

Read the English text and the German translation hypothesis and then correct the output. If the hypothesis is already correct, do not make any changes.

With this prompt, the models were given triples of (source, hypothesis, reference) with the hypothesis from our annotated data set and the reference coming from the parallel data.

In the *MRK* scenario, models were prompted to

Read the English text and the German translation hypothesis and then correct the output. Incorrect words are inside of tags '<bad> </bad>'. Please use this feedback in your correction. If the hypothesis is already correct, do not make any changes.

As with the *APE* prompt, models were given (source, hypothesis, reference) triples with the tokens that were marked as bad during annotation inside of XML-style tags, <bad></bad>. We decided that giving the error markings as in-line tags would be easier for the model to parse and integrate in its output than including another line where errors would be indicated further away from the corresponding tokens.

---

[3]We used the model *all-MiniLM-L6-v2* and retrieved the examples with the highest cosine similarity.

Condition	BLEU	TER	ME	UE	% Correct ME
Original Hyps	28.92	55.12	N.A.	N.A.	N.A.
MT (Llama/GPT)	29.83/38.61	55.97/49.21	N.A.	N.A.	N.A.
APE (Llama/GPT)	29.79/39.09	54.56/48.37	7.30/76.70	1.76/15.85	32% / N.A.
MRK (Llama/GPT)	30.09/39.31	54.70/48.32	14.76/78.36	3.60/13.90	67% / N.A.

Table 5: Results for both Llama 13B and GPT 3.5 across all metrics and translation scenarios (ME = Marking Edits, UE = Unmarking Edits, % Correct ME = Percentage of correct ME in manual evaluation).

5.2 Metrics

We evaluate the models' new hypotheses with a suite of metrics to check for token level matches, semantic similarity, and error marking usage. We use the token based metrics BLEU[4] (Papineni et al., 2002) and TER[5] (Snover et al., 2006) as implemented in SacreBLEU (Post, 2018). We did not include the popular neural metric COMET (Rei et al., 2020) since is not sensitive to the individual token changes (Glushkova et al., 2023) that we ask the LLMs to perform.

In addition to these metrics, we also implement our own to see how well the models are at recognizing and making edits to errors. These are called *marking edit* (ME) and *unmarking edit* (UE). We perform a word-level diff in order to see which words need to be edited or deleted in the original hypothesis to arrive at the new hypothesis. Combining this with the error markings allows us to examine if edits were made to the tokens error-marked by the annotators (ME), or if tokens that were otherwise OK were changed (UE).

The ME and UE metrics, however, cannot tell if the edits were correct, only that they were made. To determine if the edits correctly fix the errors, we performed a manual evaluation on the marking edits produced by Llama 13B in both the *APE* and *MRK* settings. Three of the authors contributed to the evaluation. Two are native speakers of German and fluent in English while one is a native speaker of English and fluent in German. We selected 100 sentences with the most marking edits. Edits were evaluated in terms of their correctness, with a subjective yes or no answer given for the entire sentence.

6 Results

We show results across metrics for Llama 13B and GPT-3.5 in Table 5. Including error markings as input increases the frequency with which the models edits the marked tokens. For Llama 13B, we see editing rates for marked tokens double from 7.30 to 14.76. This suggests that, even after being asked to correct the hypotheses, Llama 13B finds its own outputs as acceptable translations. When errors are specifically pointed out to the model, it is much more capable of self-correcting errors.

Llama 13B nominally improves BLEU scores over the original hypotheses score (28.92) in all scenarios with *MRK* in the lead with 30.09, *MT* in second with 29.83[6] and *APE* with 29.79. Nominal improvements over the original hypotheses are also found according to the TER metric, albeit only for *APE* and *MRK* scenarios.

The GPT model is already quite capable of finding errors in the hypotheses without error markings and the *APE* outputs achieve marking edits of 76.70 while *MRK* has a slight improvement of 78.36. Worth noting is the reduction in unmarking edits when prompting GPT with *MRK*. *MRK* reduces unmarking edits to 13.90 from 15.85 with *APE*. This means that indicating specific errors can constrain the number of edits that the GPT model makes. Additionally, nominal improvements of BLEU and TER scores are found in the *APE* and *MRK* scenarios over *MT* with GPT 3.5 as well. *MRK* improves BLEU to 39.31 from *MT*'s 38.61.

In the manual evaluation of marking edits, we found that *APE* made correct edits 32% of the time on average, while either making incorrect edits or not editing the rest. *MRK* on the other hand was judged correct 67% of the time on average. Agreement in terms of Krippendorff's Alpha for sentence

---

[4]BLEU signature: `nrefs:1 | case:mixed | eff:no | tok:13a | smooth:exp | version:2.4.0`
[5]TER Signature `nrefs:1 | case:1c | tok:tercom | norm:no | punct:yes | asian:no | version:2.4.0`

[6]*MT* is able to surpass the original hypotheses with Llama 13B because the annotated hypotheses were generated with the same 5 examples for all inference segments while *MT* retrieved similar examples for each test segment.

level ratings of *APE* is $\alpha = 0.82$, while for *MRK* $\alpha = 0.55$. As *APE* makes fewer edits overall, it is easier to classify as incorrect or not editing. For *MRK* there was disagreement on how to handle partial edits or if not all markings were edited, requiring individual judgement by each evaluator.

7 Conclusion

We presented a pilot study to investigate the potential of augmenting a so-called PE-TM resource consisting of sources, machine translations, and human references, with human error markings in order to guide an LLM to self-correct marked erroneous term translations. We find that the LLM that produced the translation hypotheses identifies its own translations as correct, and therefore does not act on the instructions to correct errors. However, when prompted with error markings, the LLM learns to act on them, doubling the number of edits to marked tokens, with nearly 70% of the edits being correct according to a human evaluation. In sum, our pilot study shows that the additional effort of error marking a machine translation at test time allows an LLM translation system to learn focused corrections on marked errors from similar examples extracted from a PE-TM, leading to improved translation quality over APE and MT. In future work, we will investigate learned models for error markings. These require larger TMs for reliable training of markings estimators, but also bear the promise of improved retrieval augmentation.

8 Acknowledgements

References

Berger, Nathaniel, Miriam Exel, Matthias Huck, and Stefan Riezler. 2023. Enhancing supervised learning with contrastive markings in neural machine translation training. In *Proceedings of the 24th Annual Conference of The European Association for Machine Translation (EAMT)*, Tampere, Finland.

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

Exel, Miriam, Bianka Buschbeck, Lauritz Brandt, and Simona Doneva. 2020. Terminology-constrained neural machine translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisboa, Portugal.

Garcia, Xavier, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, Hanolulu, Hawaii, USA.

Ghazvininejad, Marjan, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *arXiv*, abs/2302.07856.

Glushkova, Taisiya, Chrysoula Zerva, and André F. T. Martins. 2023. BLEU meets COMET: Combining lexical and neural metrics towards robust machine translation evaluation. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation (EACL)*, Tampere, Finland.

Grangier, David and Michael Auli. 2018. QuickEdit: Editing text & translations by crossing words out. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL:HLT)*, New Orleans, Louisiana.

Hoang, Cuong, Devendra Sachan, Prashant Mathur, Brian Thompson, and Marcello Federico. 2023. Improving retrieval augmented neural machine translation by controlling source and fuzzy-match interactions. In *Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia.

Huang, Jie, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv*, abs/2310.01798.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain.

Kreutzer, Julia, Nathaniel Berger, and Stefan Riezler. 2020. Correct me if you can: Learning from error corrections and markings. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Krippendorff, Klaus. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research*, 30(3):411–433.

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

Madaan, Aman, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv*, abs/2303.17651.

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

Moslem, Yasmin, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation (EAMT)*, Tampere, Finland.

OpenAI. 2023. GPT-4 technical report. Technical report, OpenAI.

Pan, Liangming, 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*, abs/2308.03188.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, Philadelphia, PA.

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

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium.

Raunak, Vikas, Amr Sharaf, Yiren Wang, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong King, China.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA'06)*, Cambridge, MA.

Tiedemann, Jörg. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv*, abs/2302.13971.

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

Vidal, Blanca, Albert Llorens, and Juan Alonso. 2022. Automatic post-editing of MT output using large language models. In Campbell, Janice, Stephen Larocca, Jay Marciano, Konstantin Savenkov, and Alex Yanishevsky, editors, *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, Orlando, Florida, USA.

Vilar, David, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In Rogers, Anna, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, Toronto, Canada.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online.

Xu, Wenda, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2023. Pinpoint, not criticize: Refining large language models via fine-grained actionable feedback.

A Appendix

A.1 Example Prompts

An example of a prompt for *MT*, *APE*, and *MRK* are in Figures 3, 4, and 5, respectively.

Translate English to German.

English: Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser.

German: Sie sind Bestandteil des HTTP-Headers, was bedeutet, dass die Funktion setcookie() aufgerufen werden muss, bevor irgendeine Ausgabe an den Browser erfolgt.

English: session.use_only_cookies specifies whether the module will only use cookies to store the session id on the client side.

German: session.use_only_cookies spezifiziert, ob das Modul nur Cookies verwendet, um die Session-ID clientseitig zu speichern.

English: Note that SID is only defined if the client didn't send the right cookie.

German: Beachten Sie, dass SID nur definiert ist, wenn vom Client nicht das richtige Cookie gesendet wurde.

English: The server does not support the request type of the body.

German: Der Server unterstützt den angeforderten Typ nicht.%1: request type

English: Must be in active session on local console

German: Nur in aktiver Sitzung auf lokaler Konsole

English: Like other headers, cookies must be sent before any output from your script (this is a protocol restriction).

German:

Figure 3: Example of 5-shot prompt for English-to-German Translation. Each demonstration example consists of a source segment in English (in green), and a reference translation (in red).

Read the English text and the German translation hypothesis and then correct the output. If the hypothesis is already correct, do not make any changes.

English: Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser.
Hypothesis: Cookies sind Teil des HTTP-Headers , deshalb muss setcookie() vor jedem Ausgabe-Output an den Browser aufgerufen werden .
German: Sie sind Bestandteil des HTTP-Headers, was bedeutet, dass die Funktion setcookie() aufgerufen werden muss, bevor irgendeine Ausgabe an den Browser erfolgt.

English: session.use_only_cookies specifies whether the module will only use cookies to store the session id on the client side.
Hypothesis: session .use_only_cookies bestimmt , ob das Modul nur mit Cookies die Session-ID auf dem Client-Betriebssystem speichert .
German: session.use_only_cookies spezifiziert, ob das Modul nur Cookies verwendet, um die Session-ID clientseitig zu speichern.

English: Note that SID is only defined if the client didn't send the right cookie.
Hypothesis: Beachtet , dass SID nur definiert ist , wenn der Client nicht den richtigen Cookie gesendet hat .
German: Beachten Sie, dass SID nur definiert ist, wenn vom Client nicht das richtige Cookie gesendet wurde.

English: The server does not support the request type of the body.
Hypothesis: Der Server unterstützt nicht die Anforderungstyp der Body .
German: Der Server unterstützt den angeforderten Typ nicht.%1: request type

English: Must be in & active session on local console
Hypothesis: Muss in & aktiver Sitzung auf dem lokalen Konsole
German: Nur in & aktiver Sitzung auf lokaler Konsole

English: Like other headers, cookies must be sent before any output from your script (this is a protocol restriction).
Hypothesis: Wie andere Headern müssen Cookies vor jedem Ausgabe-Output (dies ist eine Protokoll-Einschränkung) gesendet werden .
German:

Figure 4: Example of 5-shot prompt for English-to-German Automatic Post-Editing (*APE*). Each demonstration example consists of a source segment in English (in green), a translation hypothesis in German (in blue), and a reference translation (in red).

Read the English text and the German translation hypothesis and then correct the output. Incorrect words are inside of tags '<bad> </bad>'. Please use this feedback in your correction. If the hypothesis is already correct, do not make any changes.

English: Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser.
Hypothesis: Cookies sind Teil des HTTP-Headers , deshalb muss setcookie() vor jedem <bad> Ausgabe-Output </bad> an den Browser aufgerufen werden .
German: Sie sind Bestandteil des HTTP-Headers, was bedeutet, dass die Funktion setcookie() aufgerufen werden muss, bevor irgendeine Ausgabe an den Browser erfolgt.

English: session.use_only_cookies specifies whether the module will only use cookies to store the session id on the client side.
Hypothesis: session .use_only_cookies bestimmt , ob das Modul nur <bad> mit </bad> Cookies die Session-ID auf dem <bad> Client-Betriebssystem speichert </bad> .
German: session.use_only_cookies spezifiziert, ob das Modul nur Cookies verwendet, um die Session-ID clientseitig zu speichern.

English: Note that SID is only defined if the client didn't send the right cookie.
Hypothesis: <bad> Beachtet </bad> , dass SID nur definiert <bad> ist </bad> , wenn der Client nicht den richtigen Cookie gesendet hat .
German: Beachten Sie, dass SID nur definiert ist, wenn vom Client nicht das richtige Cookie gesendet wurde.

English: The server does not support the request type of the body.
Hypothesis: Der Server unterstützt nicht <bad> die </bad> Anforderungstyp <bad> der </bad> Body .
German: Der Server unterstützt den angeforderten Typ nicht.%1: request type

English: Must be in & active session on local console
Hypothesis: Muss in & aktiver Sitzung auf <bad> dem </bad> lokalen Konsole
German: Nur in & aktiver Sitzung auf lokaler Konsole

English: Like other headers, cookies must be sent before any output from your script (this is a protocol restriction).
Hypothesis: Wie andere <bad> Headern </bad> müssen Cookies vor jedem <bad> Ausgabe-Output </bad> (dies ist eine Protokoll-Einschränkung) gesendet werden .
German:

Figure 5: Example of 5-shot prompt for English-to-German Post-Editing with error markings (*MRK*). Error markings inside by tags <bad> </bad>. Each demonstration example consists of a source segment in English (in green), a translation hypothesis in German (in blue), and a reference translation (in red).