KU X Upstage's submission for the WMT22 Quality Estimation: Critical Error Detection Shared Task

Sugyeong Eo[1], **Chanjun Park**[1,2], **Hyeonseok Moon**[1], **Jaehyung Seo**[1], **Heuiseok Lim**[1*]
[1]Korea University [2]Upstage
{djtnrud,bcj1210,glee889,seojae777,limhseok}@korea.ac.kr
chanjun.park@upstage.ai

Abstract

This paper presents KU X Upstage's submission to the quality estimation (QE): critical error detection (CED) shared task in WMT22. We leverage the XLM-RoBERTa large model without utilizing any additional parallel data. To the best of our knowledge, we apply prompt-based fine-tuning to the QE task for the first time. To maximize the model's language understanding capability, we reformulate the CED task to be similar to the masked language model objective, which is a pre-training strategy of the language model. We design intuitive templates and label words, and include auxiliary descriptions such as demonstration or Google Translate results in the input sequence. We further improve the performance through the template ensemble, and as a result of the shared task, our approach achieve the best performance for both English-German and Portuguese-English language pairs in an unconstrained setting.

1 Introduction

This paper presents our submission to the critical error detection (CED) shared task among the quality estimation (QE) tasks of WMT22 (Zerva et al., 2022). CED is a task of detecting cases where translation errors in source sentences or translation results distort meaning in terms of race, gender, safety, law, finance, etc. (Specia et al., 2021; Rubino et al., 2021; Jiang et al., 2021). Critical translation errors in the shared task appear in the form of mistranslation, hallucination, and deletion in source sentences or translation results, and errors can be classified into five categories: additions, deletions, named entities, meaning, and numbers. Even if machine translation (MT) systems produce fluent translations, the fact that they cannot be free from fatal semantic errors emphasizes the importance of preventing social repercussions from the errors. Forbidding socially bad influences and losses from these meaning deviations is the purpose of the CED task (Specia et al., 2021).

Participating systems distinguish only critical errors, not correct translations or simple translation errors. In contrast to last year, submissions should be provided with continuous scores rather than binary labels. The official script calculates scores with automatically assigned classes based on the threshold value of the index corresponding to the number of errors. Similar to last year, we participated in unconstrained English-German (En-De) and Portuguese-English (Pt-En) utilizing released training datasets[1].

To perform the CED task, we exploit the XLM-RoBERTa large model (Conneau et al., 2019) as utilized in the baseline without additional parallel data. In addition, we adopt prompt-based fine-tuning to mitigate catastrophic forgetting during fine-tuning by maximizing the linguistic capability obtained through pre-training. In prompt-based fine-tuning, the downstream task is reformulated into a cloze-style, which is consistent with the masked language modeling objective. The word for the masked part is predicted by the model based on the task-specific template (Liu et al., 2021a). Recent studies have demonstrated the remarkable effects of prompt-based learning in the natural language processing field (Brown et al., 2020; Gao et al., 2020; Schick and Schütze, 2020; Liu et al., 2021b; Zhao and Schütze, 2021), and we apply this new paradigm to the QE task. We manually generate templates each containing a source sentence, its translation result, and a description with a mask token for the CED task. Furthermore, we generate label words (Liu et al., 2021b) to map the words to be filled in the masked part and labels.

Exploring appropriate templates in prompt-based fine-tuning is important because the performance ranges widely depending on the template

* Corresponding Author

[1]The following is the leaderboard of the CED task. https://codalab.lisn.upsaclay.fr/competitions/6893

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 606–614
Abu Dhabi, December 7–8, 2022.

used. Therefore, we design multiple hard prompts through prompt engineering, and these are configured into three types of templates according to additional information: plain template, template with demonstration, and template with Google Translate. Through answer engineering, we map contrastive words for each OK and BAD tag in diverse combinations. To obtain the final score, we extract probability for words mapped to BAD. We further improve performance by ensembling values from templates.

Our approach outperforms the baseline models in En-De and Pt-En by a substantial margin and achieves first place. Experimental results demonstrate that simply setting up the training method without modifying the model or augmenting the data with additional parallel corpora significantly affects the performance.

Template
<s> **src** </s> **mt**. <mask> translation.</s>
<s> **src** </s> **mt**. It was <mask> translation.</s>
<s> A <mask> translation of **src** is **mt**.</s>
<s> **src** </s> **mt** <mask></s>
<s> **src** </s> **mt**? <mask></s>
<s> **src** </s> **mt**? <mask>,</s>
<s> **src** </s> **mt**? "<mask>"</s>
Label Words
OK: "great", BAD: "terrible"
OK: "good", BAD: "bad"
OK: "!", BAD: "?"
OK: "nice", BAD: "poor"
OK: "yes", BAD: "no"

Table 1: Prompt templates and label words utilized in our experiments. We denote a source sentence as **src** and its translation result as **mt**.

2 Proposed Method

2.1 Prompt-based Fine-tuning

We adopt prompt-based fine-tuning to diminish the discrepancy between the training objectives of the fine-tuning and pre-training (Shin et al., 2020). By applying this, we induce our CED model to preserve the linguistic capability obtained via the pre-training phase.

In our task, we denote $(src, mt, y) \in D$ for a CED training dataset D, where src and mt denote a source sentence and its translated sentence, respectively, and y denotes its corresponding label (*e.g.* OK, BAD). Furthermore, we define two mapping functions T, L that transform all the data in D to implement prompt-based fine-tuning in the CED task.

The template function T transforms each src and mt into a single input sequence that contains description with masked token. In generating the input sequence, T also defines the placement of a special <mask> token to fill in. During training, we induce the model to infer the appropriate word suitable for the corresponding <mask> token position that is coherent with the overall context. Subsequently, the label word function, referred to as verbalizer, L transforms the given label y into an appropriate label word to be placed in the masked position of the input sequence transformed through T.

For example, given src as "indigenous peoples constitute just 0.7% of the global population", mt as "Indigene Völker machen nur 5% der Weltbevölkerung aus", and their corresponding label y as BAD with label words "OK:great, BAD:terrible", T convert these sentences into "<s> indigenous peoples constitute just 0.7% of the global population </s> Indigene Völker machen nur 5% der Weltbevölkerung aus </s>. It was <mask> translation." and L convert its label into "terrible". Then the original fine-tuning objective of CED that determines whether the label is "OK" or "BAD" is converted to predict the correct word for the <mask> token position. Specifically, the model is trained to predict the following probability:

$$P(y|src, mt) = P(\langle mask \rangle = L(y)|T(src, mt)) \quad (1)$$

Considering the scoring method of the WMT22 CED task, we do not binarize the model inference results into a OK or a BAD tag. Instead, we use the softmax function to normalize the overall score as in Equation (2) and extract the probability that the decoded token in the <mask> position will be mapped to BAD. We regard this probability as the estimated quality score of the mt.

$$\text{score}(src, mt) = \frac{\exp(P(\texttt{BAD}|src, mt))}{\sum_{y' \in \{\texttt{OK},\texttt{BAD}\}} \exp(P(y'|src, mt))} \quad (2)$$

2.2 Prompt and Answer Engineering

Because the effective prompt for the CED task has not been revealed, we design various prompt candidates (Gao et al., 2020). We attempt to organize the model input into a natural context, such as "src

	En-De			Pt-En		
	Train	Dev	Test	Train	Dev	Test
# of Sentences	155511	17280	500	39925	4437	500
Avg **src** Toks	22.98	23.07	24.15	25.49	25.5	26.63
Avg **mt** Toks	23.71	23.8	24.68	22.52	22.39	23.26
Min/Max **src** Toks	2/112	2/90	4/82	2/117	2/85	3/74
Min/Max **mt** Toks	2/106	2/109	4/80	1/107	2/82	3/69
% of **BAD** label	6.1	5.82	-	6.05	5.79	-

Table 2: Dataset statistics on WMT22 CED task

mt. It was <mask> translation, A <mask> translation of src is mt". For the label words, we select two distinct words, such as "great/terrible", and "good/bad". We intend to obviate ambiguity during model training by establishing clear contrasting label words, although naive errors are not considered a good translation result. All types of templates and label words are listed in Table 1, and the entire prompt used in our experiments is described in Appendix A.

2.3 Auxiliary Description

We append auxiliary descriptions that provide supplementary information to the model input (Gao et al., 2020; Chen et al., 2021; Brown et al., 2020). We select two types of auxiliary descriptions: demonstration and Google Translate results.

The demonstration extracts a single example for each class from the training data and concatenates them into the input sequence, similar to the in-context learning approach proposed in GPT3 (Brown et al., 2020) and LM-BFF (Gao et al., 2020). In contrast to LM-BFF, we randomly select training examples without any constraints on sampling to avoid unintended bias that may occur when extracting demonstrations based on semantic similarity.

The Google Translate results append translation results from the commercialized MT system. As demonstrated in previous studies (Chen et al., 2021; Wang et al., 2020; Moon et al., 2021), adding Google Translate results contributes to a significant performance improvement. Regarding this, we use Google Translate to generate mt' by translating each src in the entire data. By adding this to the input sequence, we distill the knowledge of the external MT system into the model.

Auxiliary descriptions are combined with each example in D to compose a new input sequence. Through this, we induce the model to determine the critical errors by grounding more information.

2.4 Prompt Ensemble

As mentioned previously, prompt-based fine-tuning shows various deviations in model performance depending on the designed prompts (Shin et al., 2020). We aim to boost performance by aggregating the results from multiple prompts to minimize bias and distribute contributions per template. For the ensemble, we add the top K values with high Matthew's correlation coefficient (MCC) results.

3 Experimental Setting

3.1 Dataset Details

We leverage the dataset provided by WMT22[2]. The dataset statistics for each language pair are reported in Table 2. In summary, a sentence contains an average of 22 to 26 tokens, with a bad tag ratio of 5-6%. When using auxiliary descriptions, we randomly extract data corresponding to `OK` and `BAD` tags from the training dataset to configure the demonstration. When leveraging the commercialized MT result, we translate source sentences using the most widely adopted Google Translate[3].

We tokenize sentences with the XLM-RoBERTa tokenizer. Considering the average token and maximum sequence length of statistics, after concatenating src and mt, we filter cases where the tokenized sentence length is over 250. We score our predictions with the official script[4] provided by WMT22 and MCC.

3.2 Model Details

We exploit the same multilingual language model, XLM-RoBERTa large (Conneau et al., 2019), for both En-De and Pt-En language pairs and leverage the model and tokenizer[5] distributed by Huggingface (Wolf et al., 2019). For conducting prompt-based fine-tuning, we experiment after modifying LM-BFF[6] framework. In the case of hyperparameters, the max sequence length is set to 256 and batch size is set to 32 if auxiliary description is not used in model training, otherwise we set the max sequence length to 350 and batch size to 16. As shown in Table 2, considering the total data size for each language pair, we train Pt-En to 10K training

[2] https://github.com/WMT-QE-Task/wmt-qe-2022-data

[3] https://translate.google.co.kr/

[4] https://github.com/WMT-QE-Task/wmt-qe-2022-data/blob/main/critical-errors-subtask/official_evaluation.py

[5] xlm-roberta-large

[6] https://github.com/princeton-nlp/LM-BFF.git

	En-De			Pt-En		
	MCC (Binary)	**MCC**	**P&R**	**MCC (Binary)**	**MCC**	**P&R**
Baseline	-	0.8943	0.9001	-	0.8955	0.9012
Plain (Avg)	**0.9161 ±0.0037**	**0.9117 ±0.0075**	**0.9166 ±0.0071**	0.9223 ±0.0089	0.9042 ±0.0217	0.9095 ±0.0206
Demo (Avg)	0.9121 ±0.0062	0.9072 ±0.0115	0.9123 ±0.0109	0.9118 ±0.0113	0.9003 ±0.0266	0.9053 ±0.0246
Google MT (Avg)	0.9143 ±0.0272	0.9092 ±0.0217	0.9142 ±0.0205	**0.9391 ±0.0331**	**0.9312 ±0.0444**	**0.9350 ±0.0238**
Plain (Max)	0.9189	0.9153	0.9200	0.9312	0.9173	0.9218
Demo (Max)	0.9183	**0.9187**	0.9160	0.9231	0.9173	0.9177
Google MT (Max)	**0.9218**	0.9165	**0.9211**	**0.9649**	**0.9565**	**0.9588**

Table 3: Unconstrained En-De, Pt-En development (dev) set result on WMT22 CED task. We measure MCC from the WMT22 official script. As the official script refines the inference results to have the same distribution of OK and BAD with reference, precision and recall always indicate the same value. Therefore, we denote precision and recall as P&R. We further present the MCC (Binary) result measured through the binary label. This result tends to be higher than the official script.

	En-De		Pt-En	
	MCC	**P&R**	**MCC**	**P&R**
All	0.9265	0.9305	0.9565	0.9588
Top 5	0.9309	0.9347	0.9695	0.9712
Top 10	0.9309	0.9347	**0.9739**	**0.9753**
Top 15	**0.9321**	**0.9358**	0.9652	0.9671
Truncate	0.9287	0.9326	0.9521	0.9547

Table 4: MCC results on top K template ensemble. Truncate indicates an ensemble result only when the dev MCC is over the baseline result.

steps and En-De to 35K. As a GPU setting, one RTX 8000 is used for learning.

4 Experimental Results

4.1 Prompt-based Fine-tuning Results

We present the prompt-based fine-tuning results for En-De and Pt-En language pairs in Table 3. We mainly divide results into three categories: plain template, template with demo, and template with Google Translate according to the auxiliary description we used. Each consists of 8, 11, and 20 different templates, and we report the average and max values in the table. Performances by leveraging each template is described in Appendix A. The baseline is the official fine-tuning results for the XLM-RoBERTa large model. Our approach significantly outperforms the baseline performance in average and maximum performance for all experiments.

Specifically, templates with no auxiliary description (*i.e.* Plain) show comparatively high results in En-De. When using demonstration (*i.e.* Demo) and Google Translate (*i.e.* Google MT), the performance is slightly decreased. However, templates with a demonstration show effective benefits in the max MCC. In addition, the best performance is achieved in templates with Google Translate in the case of MCC (Binary), which measured MCC by comparing binary predictions and labels. Through the results, we conclude that including additional information in the input sequence leads to performance improvement.

Pt-En Google MT MCC results strongly support our hypothesis. Additional translation results within the input sequence competitively contribute to performance improvement in both average and max, outperforming +0.0392 MCC over the Plain (Max). When comparing demonstration and Google Translate, we infer that presenting information related to the input example has a better effect on learning than providing representative examples of tasks.

In the average results (*i.e.* Avg), the performance gap per template is indicated by ±. Under the setting where the selected auxiliary description is fixed, the performance of different templates varies considerably, from 0.0037 to 0.0217 MCC for En-De and from 0.0089 to 0.444 MCC for Pt-En. Therefore, we perform ensembles to obtain the final score by aggregating the top K predictions.

4.2 Template Ensemble Results

Table 4 is the ensemble results of the top K templates, showing notable performance. Ensembles against the top 15 templates for En-De and the top 10 templates for Pt-En yield the best MCC results. These show +0.0134 MCC higher in En-De and +0.0174 MCC higher in Pt-En than the max results listed in Table 3. This demonstrates that the distributed contribution to multiple prompts per example further improves the final performance.

	En-De		Pt-En	
	MCC	**P&R**	**MCC**	**P&R**
Baseline	0.855	0.873	0.934	0.944
aiXplain	0.219	0.318	0.179	0.296
Ours	**0.964**	**0.968**	**0.984**	**0.986**

Table 5: Official result on En-De, Pt-En CED blind test set

Furthermore, we note All and Truncate in the table. The former ensembles all results and the latter removes templates with lower results than the baseline evaluation MCC before ensembling. Through this, we observe that including most of the templates does not necessarily contribute to performance improvement. High performance is obtained by training models with various types of templates and selecting appropriate predictions together.

4.3 Results on Test dataset

The experimental results for the test set are shown in Table 5. We submit the final score obtained through the ensemble. As a result, we significantly outperform the baseline result, achieving +0.109 MCC in En-De and +0.05 MCC in Pt-En. This is a notable margin because we use the same model as the baseline without utilizing any supplementary parallel data or scaling model parameters.

5 Conclusion

We applied prompt-based learning to the CED task by forming a learning objective for the task similar to that in pre-training. This method outperformed the baseline performance while preserving the model parameters and data settings. We performed manual prompt engineering and answer engineering to explore intuitive hard prompts. In addition, because finding optimal prompts is difficult, we ensembled predictions from diverse templates to address the performance variation and achieve additional performance boost. Our method is simple but powerful, and we hope that this method will be actively introduced to QE tasks in future studies.

Limitations

This study used models trained only on English-German and Portuguese-English language pairs. Therefore, language extension is challenging because data for training the CED task must be prepared for each language pair and direction. Non-trivial costs may be incurred in the data construction process. Furthermore, because we manually generated prompts and answer engineering, finding the optimal prompt is sub-optimal. Soft prompts that leverage the trained embedding values in the prompt configuration can mitigate this limitation. However, because embedding vectors in soft prompt are not described in human words, and we are the first to introduce prompt-based learning in QE tasks, we focused on interpretability.

Ethics Statement

We created task-specific templates during prompt engineering. We have not used problematic statements at this time. In addition, when engineering label words, words without ethical issues were used. However, unethical expressions such as socially problematic words and abusive language are included in the CED data. Owing to the nature of the task, this is intentionally appended by annotators to detect critical errors. The purpose of this task is to classify and exclude ethical issues that occur in the machine translation field.

Acknowledgements

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2022-2018-0-01405) supervised by the Institute for Information & Communications Technology Planning & Evaluation(IITP) and this work was supported by an IITP grant funded by the MSIT (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and this research was supported by the MSIT, Korea, under the ICT Creative Consilience program (IITP-2022-2020-0-01819) supervised by the IITP.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, et al. 2021. Hw-tsc's participation at wmt 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Genze Jiang, Zhenhao Li, and Lucia Specia. 2021. Icl's submission to the wmt21 critical error detection shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 928–934.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Hyeonseok Moon, Chanjun Park, Sugyeong Eo, Jaehyung Seo, and Heuiseok Lim. 2021. An empirical study on automatic post editing for neural machine translation. *IEEE Access*, 9:123754–123763.

Raphaël Rubino, Atsushi Fujita, and Benjamin Marie. 2021. Nict kyoto submission for the wmt'21 quality estimation task: Multimetric multilingual pretraining for critical error detection. In *Proceedings of the Sixth Conference on Machine Translation*, pages 941–947.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André FT Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, et al. 2020. Hw-tsc's participation at wmt 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *arXiv preprint arXiv:2109.03630*.

A Appendix

A.1 Results on Each Template

The evaluation MCC results for each template of the En-De and Pt-En pairs are shown in Tables 6 and 7. Particularly in Pt-En, the performance of each template varies considerably, reporting significantly lower or superior performance than the baseline result.

Type	Index	Template	Label Words	MCC (Binary)	MCC	P&R
Plain Template	1	**src mt**. <mask>translation.	great / terrible	0.9156	**0.9153**	0.9200
	2	**src mt**. <mask>translation.	good / bad	0.9158	0.9087	0.9137
	3	**src mt** <mask>	! / ?	0.9153	**0.9131**	0.9179
	4	A <mask>translation of **src** is **mt**.	good / bad	0.9189	**0.9142**	0.9189
	5	A <mask>translation of **src** is **mt**.	great / terrible	0.9161	**0.9131**	0.9179
	6	**src mt**. It was <mask>translation.	great / terrible	0.9124	0.9042	0.9095
	7	**src mt**. It was <mask>translation.	nice / poor	0.9161	**0.9131**	0.9179
	8	**src mt**? <mask>	yes / no	0.9184	***0.9120***	0.9168
Template with Demo	1	**demo_ok demo_bad srcmt**. <mask>translation.	great / terrible	0.9149	0.9064	0.9116
	2	**demo_ok demo_bad src mt**. <mask>translation.	good / bad	0.9089	0.9098	0.9147
	3	**demo_ok demo_bad src mt**. It was <mask>translation.	great / terrible	0.9125	0.9064	0.9116
	4	**demo_ok demo_bad src mt**. It was <mask>translation.	nice / poor	0.9183	**0.9187**	0.9232
	5	**demo_ok demo_bad src mt**. It was <mask>translation.	! / ?	0.9084	0.9042	0.9095
	6	**demo_ok demo_bad src mt**? <mask>	yes / no	0.9109	0.9075	0.9126
	7	**src mt demo_ok demo_bad** <mask>translation.	great / terrible	0.9095	0.8964	0.9021
	8	**src mt demo_ok demo_bad** <mask>translation.	good / bad	0.9060	0.9042	0.9095
	9	**src mt demo_ok demo_bad** . It was <mask>translation.	great / terrible	0.9123	0.9064	0.9116
	10	**src mt demo_ok demo_bad** . It was <mask>translation.	nice / poor	0.9138	0.9098	0.9147
	11	**src mt demo_ok demo_bad** ? <mask>	yes / no	0.9175	0.9098	0.9147
Template with Google Translate	1	**src mt**? <mask>**gmt**	great / terrible	0.9161	0.9098	0.9147
	2	**src mt**? <mask>**gmt**	good / bad	0.9198	**0.9165**	0.9211
	3	**src mt**? <mask>**gmt**	! / ?	0.9218	**0.9165**	0.9211
	4	**src mt**? <mask>**gmt**	yes / no	0.9121	0.9087	0.9137
	5	**src mt**? It was <mask>. **gmt**	great / terrible	0.9173	0.9053	0.9105
	6	**src mt**? It was <mask>. **gmt**	good / bad	0.9166	0.9087	0.9137
	7	**src mt**? It was <mask>. **gmt**	! / ?	0.9172	**0.9153**	0.9200
	8	**src mt**? It was <mask>. **gmt**	yes / no	0.9158	***0.9120***	0.9168
	9	**src mt**? "<mask>", **gmt**	! / ?	0.9176	***0.9120***	0.9168
	10	**src mt**? "<mask>", **gmt**	good / bad	0.9175	0.9064	0.9116
	11	**src mt**? <mask>, **gmt**	! / ?	0.9103	0.9087	0.9137
	12	**src mt**? <mask>, **gmt**	good / bad	0.9111	0.9087	0.9137
	13	**src mt gmt**. <mask>translation.	great / terrible	0.9133	0.9009	0.9063
	14	**src mt gmt**. <mask>translation.	good / bad	0.8872	~~**0.8875**~~	0.8937
	15	**src mt gmt**. <mask>	! / ?	0.9151	0.9064	0.9116
	16	A <mask>translation of **src** is **mt gmt**.	good / bad	0.9209	**0.9165**	0.9211
	17	A <mask>translation of **src** is **mt gmt**.	great / terrible	0.9134	***0.9109***	0.9158
	18	**src mt gmt**. It was <mask>translation.	great / terrible	0.9134	0.9075	0.9126
	19	**src mt gmt**. It was <mask>translation.	nice / poor	0.9160	**0.9165**	0.9211
	20	**src mt gmt**? <mask>	yes / no	0.9147	0.9098	0.9147

Table 6: En-De results for all templates. The top five MCCs are in red bold, the top 10 MCCs are in orange bold and underlined, and the top 15 MCCs are in blue bold and italic. We indicate the MCC below the baseline as gray bold and strikeouts.

Type	Index	Template	Label Words	MCC (Binary)	MCC	P&R
Plain Template	1	**src mt**. <mask>translation.	great / terrible	0.9312	0.9129	0.9177
	2	**src mt**. <mask>translation.	good / bad	0.9203	0.9173	0.9218
	3	**src mt** <mask>	! / ?	0.9190	~~0.8825~~	0.8889
	4	A <mask>translation of **src** is **mt**.	good / bad	0.9250	0.9129	0.9177
	5	A <mask>translation of **src** is **mt**.	great / terrible	0.9246	0.9129	0.9177
	6	**src mt**. It was <mask>translation.	great / terrible	0.9170	~~0.8868~~	0.8930
	7	**src mt**. It was <mask>translation.	nice / poor	0.9264	0.9129	0.9177
	8	**src mt**? <mask>	yes / no	0.9150	~~0.8955~~	0.9012
Template with Demo	1	**demo_ok demo_bad srcmt**. <mask>translation.	great / terrible	0.9080	~~0.8825~~	0.8889
	2	**demo_ok demo_bad src mt**. <mask>translation.	good / bad	0.9201	0.8999	0.9053
	3	**demo_ok demo_bad src mt**. It was <mask>translation.	great / terrible	0.9178	0.9042	0.9095
	4	**demo_ok demo_bad src mt**. It was <mask>translation.	nice / poor	0.9112	0.9042	0.9095
	5	**demo_ok demo_bad src mt**. It was <mask>translation.	! / ?	0.9009	0.8999	0.9053
	6	**demo_ok demo_bad src mt**? <mask>	yes / no	0.9044	0.9129	0.9177
	7	**src mt demo_ok demo_bad** <mask>translation.	great / terrible	0.9131	0.9042	0.9095
	8	**src mt demo_ok demo_bad** <mask>translation.	good / bad	0.9056	~~0.8737~~	0.8807
	9	**src mt demo_ok demo_bad** . It was <mask>translation.	great / terrible	0.9199	0.9086	0.9136
	10	**src mt demo_ok demo_bad** . It was <mask>translation.	nice / poor	0.9231	***0.9173***	0.9173
	11	**src mt demo_ok demo_bad** ? <mask>	yes / no	0.9056	~~0.8955~~	0.9012
Template with Google Translate	1	**src mt**? <mask>**gmt**	great / terrible	0.9471	***0.9390***	0.9424
	2	**src mt**? <mask>**gmt**	good / bad	0.9558	0.9478	0.9506
	3	**src mt**? <mask>**gmt**	! / ?	0.9537	0.9434	0.9465
	4	**src mt**? <mask>**gmt**	yes / no	0.9580	**0.9565**	0.9588
	5	**src mt**? It was <mask>. **gmt**	great / terrible	0.9515	**0.9521**	0.9547
	6	**src mt**? It was <mask>. **gmt**	good / bad	0.9649	**0.9565**	0.9588
	7	**src mt**? It was <mask>. **gmt**	! / ?	0.9470	**0.9521**	0.9547
	8	**src mt**? It was <mask>. **gmt**	yes / no	0.9625	**0.9521**	0.9547
	9	**src mt**? "<mask>", **gmt**	! / ?	0.9430	***0.9390***	0.9424
	10	**src mt**? "<mask>", **gmt**	good / bad	0.9603	**0.9521**	0.9547
	11	**src mt**? <mask>, **gmt**	! / ?	0.9538	**0.9521**	0.9547
	12	**src mt**? <mask>, **gmt**	good / bad	0.9514	0.9478	0.9506
	13	**src mt gmt**. <mask>translation.	great / terrible	0.9252	***0.9173***	0.9218
	14	**src mt gmt**. <mask>translation.	good / bad	0.9219	0.9086	0.9136
	15	**src mt gmt**. <mask>	! / ?	0.9269	***0.9173***	0.9218
	16	A <mask>translation of **src** is **mt gmt**.	good / bad	0.9060	~~0.8912~~	0.8971
	17	A <mask>translation of **src** is **mt gmt**.	great / terrible	0.9125	~~0.8868~~	0.8930
	18	**src mt gmt**. It was <mask>translation.	great / terrible	0.9073	0.9042	0.9095
	19	**src mt gmt**. It was <mask>translation.	nice / poor	0.9185	0.9086	0.9136
	20	**src mt gmt**? <mask>	yes / no	0.9147	0.8999	0.9053

Table 7: Pt-En results on all templates. The color and the style of top K performances are equivalent to Table 6.