HIT-MI&T Lab's Submission to Eval4NLP 2023 Shared Task

Rui Zhang[*] **Fuhai Song**[*] **Hui Huang, Jinghao Yuan, Muyun Yang**[†] and **Tiejun Zhao**
Research Center on Language Technology,
School of Computer Science and Engineering,
Harbin Institute of Technology,
Harbin, China
{23S003048, 23S103157, huanghui, 7203610706}@stu.hit.edu.cn,
{yangmuyun, tjzhao}@hit.edu.cn

Abstract

Recently, Large Language Models (LLMs) have boosted the research in natural language processing and shown impressive capabilities across numerous domains, including machine translation evaluation. This paper presents our methods developed for the machine translation evaluation sub-task of the Eval4NLP 2023 Shared Task. Based on the provided LLMs, we propose a generation-based method as well as a probability-based method to perform evaluation, explore different strategies when selecting the demonstrations for in-context learning, and try different ensemble methods to further improve the evaluation accuracy. The experiment results on the development set and test set demonstrate the effectiveness of our proposed method.

1 Introduction

As the output quality of the machine translation systems has been improved, the evaluation of translation outputs has become more challenging and critical. On one hand, human evaluations of these outputs are often time-consuming and laborious; On the other hand, previous automatic metrics such as BLEU (Papineni et al., 2002) are becoming less reliable with little remaining correlation with human judgments (Freitag et al., 2022). As a result, the demand for next generation of automatic evaluation is stronger than ever.

Large language models (LLMs), especially Generative Pre-trained Transformer (GPT) models (Radford et al., 2019; Brown et al., 2020), have led to a revolution of research in natural language processing, including machine translation evaluation. Metrics like GEMBA (Kocmi and Federmann, 2023) explore the prompting of GPT models like ChatGPT (OpenAI, 2022) and GPT4 (OpenAI, 2023) directly leveraged as metrics. Error Analysis Prompting (Lu et al., 2023) proposes to generate human-like MT evaluations with the help of LLMs by combining Chain-of-Thoughts (Wei et al., 2022) and Error Analysis (Lu et al., 2022). Besides, other work also uses LLMs to calculate the conditional probability of the generated text as the evaluation results (Fu et al., 2023; Huang et al., 2023).

This paper describes our submission to the machine translation evaluation sub-task of the Eval4NLP 2023 Shared Task (Leiter et al., 2023). Participants of this task are required to prompt the LLMs specified by the organizers as metrics for machine translation, without any fine-tuning on the selected LLM. In our work, on the basis of four LLMs provided by the organizers, we propose a generation-based method that directs the LLM to score the translated sentence directly by generation, and a probability-based method that calculate the conditional probability of the translated sentence. We also explore different demonstration selection strategies for in-context learning (Brown et al., 2020), including bucket-based selection and similarity-based selection. What's more, we try different ensemble methods, including averaging-based ensemble and multi-agent ensemble, to further improve the performance. Experiments on the development and test set shows that we obtain competitive results in this year's shared task, verifying the effectiveness of our proposed methods.

Our contributions are summarized as follows:

- We propose two methods to apply large language models on translation quality estimation, i.e. generation-based method and probability-based method.
- We investigate different demonstration selection strategies for in-context learning, including bucket-based selection and similarity-based selection.
- We examine two ensemble methods, which are averaging-based ensemble and multi-agent

[*] These authors contributed equally to this work.
[†] Corresponding author

Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems, pages 139–148
November 1, 2023 ©2023 Association for Computational Linguistics

ensemble, to further improve the evaluation performance.

2 Approach

2.1 LLMs in the Task

This year's shared task provides a list of allowed LLMs from Huggingface model hub[1]. We participate in the small model track where four models smaller than 25B parameters are available:

- **WizardLM-13B-V1.1-GPTQ**: A four-bit quantized version of WizardLM-13B-V1.1 by Xu et al. (2023). This model is chosen due to its good performance on leaderboards.
- **Nous-Hermes-13b**[2]: A model by Nous Research. This model is also chosen due to its good performance on leaderboards.
- **OpenOrca-Platypus2-13B**: A model by Lee et al. (2023). It shows strong performance on leaderboards for a 13B model and is based on LLaMA2.
- **orca_mini_v3_7b**: This model by Mathur (2023) is smaller than the others but also performs well on LLM leaderboards. It is included to accommodate for less hardware availability.

2.2 Generation-based Method

Similar to GEMBA (Kocmi and Federmann, 2023), we start by formulating the machine translation evaluation as a natural language generation problem as shown in Figure 1. We define the machine translation evaluation task with a prompt, which is a general description of the problem, and give the model source sentence and machine translated sentence (and demonstrations) as inputs. Then we can use the LLM to generate the scores of the machine translated sentences directly at inference time, without any parameter updates.

In the generation-based method, we use 4 different prompts as listed in Figure 3 in Appendix A, to ask the model to generate a score directly. One example of them is shown as follows:

```
  Score the following translation from
{source_lang} to {target_lang} with
respect to the source sentence on a
continuous scale from 0 to 100, where
a score of zero means "no meaning
preserved" and score of one hundred means
"perfect meaning and grammar".
  {source_lang} source: "{source}"
  {target_lang} translation: "{target}"
  Score(0-100): (Score)
```

2.3 Probability-based Method

Inspired by GPTScore (Fu et al., 2023), we further explore a probability-based method, as shown in Figure 2. The core idea of this method is that when instructed to perform generation, the generative pre-trained model will assign higher probabilities to a high-quality text, and vice versa. Suppose that the machine translated sentence is $\boldsymbol{h} = \{h_1, h_2, ... , h_m\}$, then the probability-based score is defined as the logarithm sum of the following conditional probabilities:

$$score = \sum_{t=1}^{m} \log p(h_t|\boldsymbol{h}_{<t}, s, p) \qquad (1)$$

where the instruction is composed of the prompt p and the source sentence s.

In the probability-based method, we use 10 different prompts as listed in Figure 4 in Appendix A, which ask models to translate a source sentence into target language. One example of them is shown as follows:

```
  Translate the following {source_lang}
sentence into {target_lang}.
  {source_lang} source: "{source}"
  {target_lang} translation: "{target}"
```

2.4 Demonstration Selection

A surprising emergent capability of LLMs is their ability to improve on prompting-based tasks by including a very small amount of demonstrations as part of the prompt, known as in-context learning (ICL) (Brown et al., 2020). We also investigate the impact of ICL on LLMs' ability to measure translation quality.

When selecting demonstrations, we try two different strategies: bucket-based selection and similarity-based selection. The details of these two strategies are as follows:

[1] https://huggingface.co/models

[2] https://huggingface.co/NousResearch/Nous-Hermes-13b

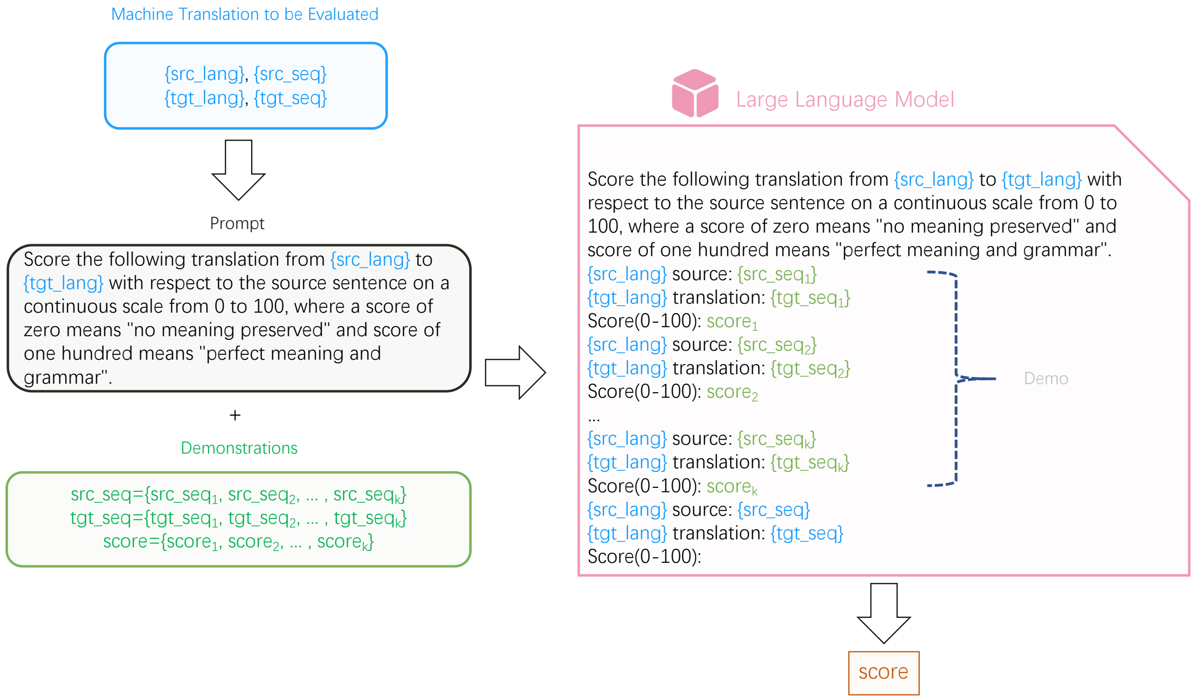
Machine Translation to be Evaluated
{src_lang}, {src_seq}
{tgt_lang}, {tgt_seq}
Prompt
Score the following translation from {src_lang} to {tgt_lang} with respect to the source sentence on a continuous scale from 0 to 100, where a score of zero means "no meaning preserved" and score of one hundred means "perfect meaning and grammar".
+
Demonstrations
src_seq={src_seq$_1$, src_seq$_2$, ... , src_seq$_k$}
tgt_seq={tgt_seq$_1$, tgt_seq$_2$, ... , tgt_seq$_k$}
score={score$_1$, score$_2$, ... , score$_k$}
Large Language Model
Score the following translation from {src_lang} to {tgt_lang} with respect to the source sentence on a continuous scale from 0 to 100, where a score of zero means "no meaning preserved" and score of one hundred means "perfect meaning and grammar".
{src_lang} source: {src_seq$_1$}
{tgt_lang} translation: {tgt_seq$_1$}
Score(0-100): score$_1$
{src_lang} source: {src_seq$_2$}
{tgt_lang} translation: {tgt_seq$_2$}
Score(0-100): score$_2$
...
{src_lang} source: {src_seq$_k$}
{tgt_lang} translation: {tgt_seq$_k$}
Score(0-100): score$_k$
{src_lang} source: {src_seq}
{tgt_lang} translation: {tgt_seq}
Score(0-100):
Demo
score

Figure 1: An example of our generation-based method. We equip the sentence pair with prompt and demonstrations, then feed them to the large language model, and ask the model to generate the evaluation score directly.

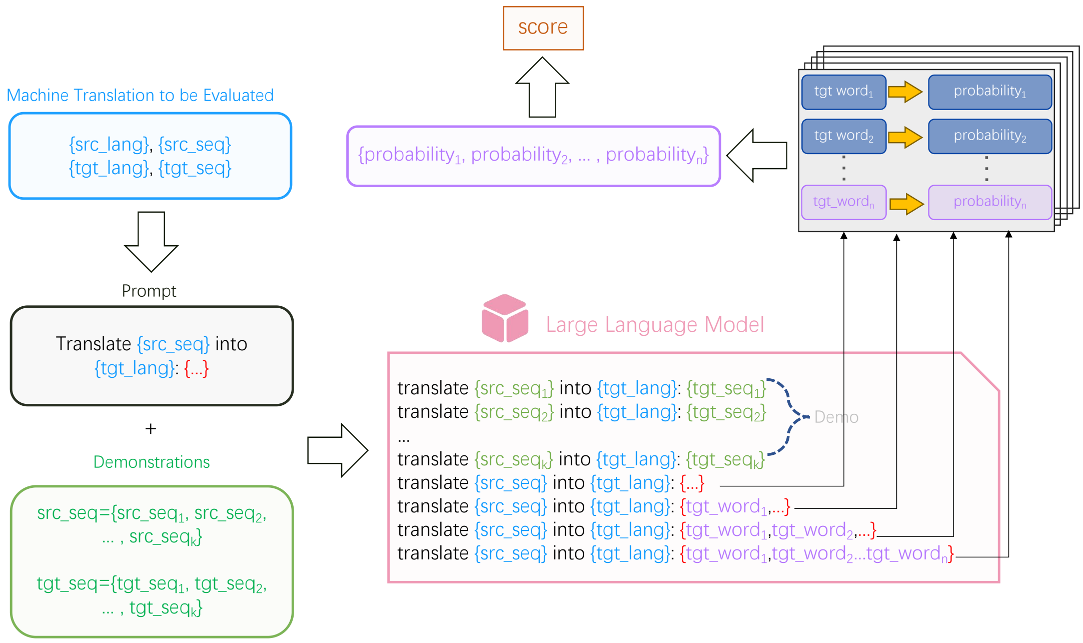
score
Machine Translation to be Evaluated
{src_lang}, {src_seq}
{tgt_lang}, {tgt_seq}
{probability$_1$, probability$_2$, ... , probability$_n$}
tgt word$_1$
probability$_1$
tgt word$_2$
probability$_2$
tgt_word$_n$
probability$_n$
Prompt
Translate {src_seq} into {tgt_lang}: {...}
+
Demonstrations
src_seq={src_seq$_1$, src_seq$_2$, ... , src_seq$_k$}
tgt_seq={tgt_seq$_1$, tgt_seq$_2$, ... , tgt_seq$_k$}
Large Language Model
translate {src_seq$_1$} into {tgt_lang}: {tgt_seq$_1$}
translate {src_seq$_2$} into {tgt_lang}: {tgt_seq$_2$}
...
translate {src_seq$_k$} into {tgt_lang}: {tgt_seq$_k$}
translate {src_seq} into {tgt_lang}: {...}
translate {src_seq} into {tgt_lang}: {tgt_word$_1$,...}
translate {src_seq} into {tgt_lang}: {tgt_word$_1$,tgt_word$_2$,...}
translate {src_seq} into {tgt_lang}: {tgt_word$_1$,tgt_word$_2$...tgt_word$_n$}
Demo

Figure 2: An example of our probability-based method. We equip the sentence pair with prompt and demonstrations, then feed them to the large language model, and calculate the conditional generation probability of every word in machine translated sentence. Then the logarithm sum of all probabilities is used as the final score.

- **bucket-based selection**: In this strategy, we first sort the candidate demonstrations according to their corresponding scores, then sequentially separate the dataset into several buckets (the number of buckets is the same as the number of demonstrations to be chosen), then we randomly choose one demonstration from every bucket.

- **similarity-based selection**: In this strategy, we select demonstrations according to their similarities to the to-be-evaluated sentence-pair. More specifically, we use two strategies to calculate the similarity of the source sentence from the dev set and the candidate demonstrations, namely BM25 (Robertson et al., 2009) and the cosine similarity of the Sentence-BERT embeddings (Reimers and Gurevych, 2019).

For generation-based method, we choose demonstrations from the training set provided by organizers. For probability-based method, we respectively choose demonstrations from De-En and En-Zh datasets of WMT newstest2020[3] for English-German (En-De) and Chinese-English (Zh-En) machine translation evaluation, and respectively choose demonstrations from De-En and Zh-En datasets of WMT newstest2020[3] and Es-En datasets of WMT newstest2012[3] for English-German (En-De), English-Chinese (En-Zh) and English-Spanish (En-Es) machine translation evaluation.

Besides, when adding demonstrations in our prompt, we also try different numbers of demonstrations, as more demonstrations might bring more reference for evaluation. We explore towards a maximum number of 10 due to length limit.

2.5 Ensemble Method

Different results from different models can be ensembled to achieve further gain. We explore two ensemble method, one is averaging-based ensemble, the other is multi-agent ensemble.

In the averaging-based ensemble, we simply calculate the average of the results of different models as the final score for each machine translated sentence.

In the multi-agent ensemble, we borrow the idea of multi-agent debate from Chan et al. (2023), where the results from different models are fed to another LLM to derive the final result. In this way, the LLM is deemed as an intelligent agent which can refer to the judgements from different models and make a final decision. The prompt we use is shown as follows:

```
  Please score the following translation
from {source_lang} to {target_lang}
with respect to the source sentence
on a continuous scale from 0 to 100,
where a score of zero means "no meaning
preserved" and score of one hundred
means "perfect meaning and grammar". As
reference, there are two other models'
scores provided.
  {source_lang} source: "{source}"
  {target_lang} translation: "{target}"
  [Score 1]: {ans1}
  [Score 2]: {ans2}
  Score(0-100): (Score)
```

Note that {ans1} is the score of the {target} provided by the first model and {ans2} is the score of the {target} provided by the second model.

3 Experiments

3.1 Set-up

Eval4NLP 2023's machine translation evaluation sub-task focuses on English-German (En-De) and Chinese-English (Zh-En) language pairs in the training and development phase. Participants are provided with a training set with 11046 En-De instances and 15750 Zh-En instances, and a development set with 7364 En-De instances and 10500 Zh-En instances. Each dataset consists of *src* (source sentence) and *mt* (machine translated sentence), and comes from MQM annotations of the WMT22 metrics shared task (Freitag et al., 2022).

In the test phase, the sub-task focuses on English-German (En-De), English-Chinese (En-Zh) and English-Spanish (En-Es) language pairs. Participants are provided with a test set with 1425 En-De instances, 1297 En-Zh instances and 1834 En-Es instances.

Kendall correlation (Kendall, 1938) is used as the evaluation metric for both two language pairs of the machine translation evaluation task.

Our experiments are all conducted on NVIDIA A800 GPU with 80G memory. The versions of pytorch and `guidance` are all the same as the versions

[3] http://www2.statmt.org/wmt23/translation-task.html#dev

Model	Demo	En-De	Zh-En
wz	0	0.0559	0.2444
wz	1	0.0963	0.2200
wz	3	**0.1404**	0.1760
wz	5	0.1103	0.1163
wz	10	0.1083	-
nh	0	0.0310	0.1995
nh	1	0.0991	0.2088
nh	3	0.1258	0.1886
nh	5	0.1355	0.1375
nh	10	0.1245	-
op	0	0.1052	**0.2502**
op	1	0.1027	0.2270
op	3	0.0659	0.1088
op	5	0.0051	0.0210
op	10	-0.0200	-0.0729
om	0	0.0453	0.1228
om	1	0.1004	0.1806
om	3	0.0636	0.1054
om	5	0.0692	0.0892
om	10	0.0608	0.1081

Table 1: Results of generation-based method on the development set with different LLMs and demonstrations. Note that "wz", "nh", "op" and "om" stand for WizardLM-13B-V1.1-GPTQ, Nous-Hermes-13b, OpenOrca-Platypus2-13B and orca_mini_v3_7b. "-" means no results due to the max length limitation of the prompt and demonstrations.

Model	Prompt	Demo	En-De	Zh-En
wz	p1	1	0.0963	0.2200
wz	p1	3	0.1404	0.1760
wz	p2	1	0.1572	**0.2283**
wz	p2	3	0.0855	0.1473
nh	p1	1	0.0991	0.2088
nh	p1	3	0.1258	0.1886
nh	p2	1	**0.1838**	0.2196
nh	p2	3	0.1419	0.1639
op	p1	1	0.1027	0.2270
op	p1	3	0.0659	0.1088
op	p2	1	0.1227	0.1906
op	p2	3	0.0170	0.0687
wz	p3	1	0.0375	0.1759
wz	p3	3	0.1043	0.1216
wz	p4	1	0.1454	0.2036
wz	p4	3	0.1142	0.1418
nh	p3	1	0.1166	0.1599
nh	p3	3	0.0612	0.0272
nh	p4	1	0.1541	0.1996
nh	p4	3	0.1200	0.1451
op	p3	1	0.0811	0.1469
op	p3	3	-0.0027	-0.0029
op	p4	1	0.1182	0.1537
op	p4	3	0.0688	0.1230

Table 2: Results of generation-based method on the development set with different LLMs, prompts and demonstrations. Note that "wz", "nh" and "op" stand for WizardLM-13B-V1.1-GPTQ, Nous-Hermes-13b and OpenOrca-Platypus2-13B. "p1", "p2", "p3" and "p4" stand for prompt 1, prompt 2, prompt 3 and prompt 4 shown in Figure 3.

Model	Strategy	Demo	En-De	Zh-En
wz	bucket	1	0.2223	0.2947
wz	bucket	3	0.2310	0.2930
wz	BM25	1	0.2165	0.2950
wz	BM25	3	0.2286	0.3001
wz	SBERT	1	0.2228	0.2959
wz	SBERT	3	0.2283	0.2987
nh	bucket	1	0.2157	0.2877
nh	bucket	3	0.2196	0.2847
nh	BM25	1	0.2107	0.2892
nh	BM25	3	0.2244	0.2930
nh	SBERT	1	0.2104	0.2910
nh	SBERT	3	0.2165	0.2937
op	bucket	1	0.2049	0.3047
op	bucket	3	0.2176	0.3023
op	BM25	1	0.2172	**0.3074**
op	BM25	3	**0.2352**	0.2921
op	SBERT	1	0.2060	0.3053
op	SBERT	3	0.2129	0.2967

Table 3: Results of probability-based method on the development set with different LLMs and demonstrations. Note that "wz", "nh" and "op" stand for WizardLM-13B-V1.1-GPTQ, Nous-Hermes-13b and OpenOrca-Platypus2-13B.

Method	Score 1	Score 2	En-De	Zh-En
probability-based	wz_p2	-	0.2347	0.2942
probability-based	op_p8	-	0.2405	0.3170
averaging-based ensemble	wz_p2	op_p8	0.2444	0.3092
multi-agent ensemble	wz_p2	op_p8	**0.2499**	**0.3192**

Table 4: Results of different models' ensemble on the development set. Note that "wz_p2" and "op_p8" stand for the score generated by WizardLM-13B-V1.1-GPTQ using the prompt 2 in Figure 4 and the score generated by OpenOrca-Platypus2-13B using the prompt 8 in Figure 4. The first and second lines are the results of probability-based method, which are generated by "wz_p2" and "op_p8".

provided by the organizers[4].

3.2 Results of Development Set

We first explore four LLMs' ability on the generation-based method, using the same prompt (Prompt 1 in Figure 3) and same demonstrations that are selected with bucket-based method. The results are shown in Table 1. We can see that orca_mini_v3_7b underperforms compared to the other three models, the reason may be its relatively fewer parameters. Besides, we find that the number of demonstrations is not the more the better, as more demonstrations may distract the model for instruction understanding.

We also explore four different prompts to further improve the generation-based method, which are shown in Figure 3. The results in Table 2 show that the change of prompt can sometimes improve the performance, but the same prompt may have quite different performance on different models. We think this is because different models may have different tendencies and comprehension abilities for prompts. Due to the vast amount of possible prompts, we believe too much prompt engineering is a cumbersome and ineffective choice.

We then measure three LLMs' performance on the probability-based method using the same prompt (Prompt 1 in Figure 4). The results in Table 3 show that our probability-based method can achieve significantly better performance than the generation-based method. We think this is because the three LLMs still lack ability of instruction following and number generation, but they are better at predicting the next token of the sentence based on their pre-training. As a result, they may underperform when scoring directly, but perform quite well when scoring with the conditional probabilities. Besides, as we can see, different selection strategies of demonstrations will cause different performance, but in general, the differences are not significant.

At last, we use the output scores from different models for ensemble and achieve further improvement. The results in Table 4 demonstrate that multi-agent ensemble perform better than the averaging-based ensemble. The reason is that multi-agent ensemble is an organic combination of the capabilities of different models by exploiting the LLM as an intelligent agent, while averaging-based ensemble simply take the average of different results without any integration.

3.3 Results of Test Set

In the test phase, we first use OpenOrca-Platypus2-13B with 10 different prompts shown in Figure 4 to generate 10 different scores, and each prompt is combined with 3 demonstrations chosen based on the Sentence-BERT-based selection strategy. Then we realize the demonstration number has a positive impact to the results, therefore we use OpenOrca-Platypus2-13B with three best prompts to generate another 3 different scores, where each prompt is combined with demonstrations as many as possible. After that, for each machine translated sentence in the test set, we feed 3 highest scores and 3 lowest scores mentioned above to OpenOrca-Platypus2-13B for ensemble, and achieve the final scores. The results are shown in Table 5 and on Codabench leaderboard[5] with the team name as HIT-MI&T Lab.

We also present the results of our probability-based method on the test set in Table 5. All the results are generated by OpenOrca-Platypus2-13B, but the number of demonstrations are different. We explore 1 demonstration, 3 demonstrations and demonstrations as many as possible, the results show that more demonstrations will lead to better performance.

[4] https://github.com/eval4nlp/SharedTask2023/blob/main/requirements.txt

[5] https://www.codabench.org/competitions/1359/#/results-tab

Model	Method	En-De	En-Zh	En-Es
OpenOrca	probability-based (1 demo)	0.4702	0.3132	0.3999
OpenOrca	probability-based (3 demo)	0.4792	0.3173	0.4054
OpenOrca	probability-based (max demo)	0.4879	0.3192	0.4093
OpenOrca	multi-agent ensemble	**0.4927**	**0.3230**	**0.4165**

Table 5: Results on the test set. Notice we present the best results among different prompts for each method.

4 Conclusion

In this paper, we describe our submission to Eval4NLP 2023 Shared Task. We propose to apply LLM on translation evaluation by generation-based method and probability-based method, and explore different demonstration selection strategies for in-context learning. Different emsemble methods are also used to obtain the final score.

A surprising result of our work is the remarkable performance of probability-based method, which performs better while requiring less token as input. In the future, we would explore more information from the inside of LLMs as the evidence to evaluate text quality.

Acknowledgements

This work is supported by National Key RD Program of China (2020AAA0108000), National Natural Science Foundation of China (62276077, U1908216), Key RD Program of Yunnan (202203AA080004) and Shenzhen College Stability Support Plan (No. GXWD20220811170358002).

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.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André FT Martins. 2022. Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Hui Huang, Shuangzhi Wu, Xinnian Liang, Bing Wang, Yanrui Shi, Peihao Wu, Muyun Yang, and Tiejun Zhao. 2023. Towards making the most of llm for translation quality estimation. In *Natural Language Processing and Chinese Computing*, pages 375–386, Cham. Springer Nature Switzerland.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Ariel N. Lee, Cole J. Hunter, Nataniel Ruiz, Bleys Goodson, Wing Lian, Guan Wang, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorcaplatypus: Llama2-13b model instruct-tuned on filtered openorcav1 gpt-4 dataset and merged with divergent stem and logic dataset model. https://huggingface.co/Open-Orca/OpenOrca-Platypus2-13B,.

Christoph Leiter, Juri Opitz, Daniel Deutsch, Yang Gao, Rotem Dror, and Steffen Eger. 2023. The eval4nlp 2023 shared task on prompting large language models as explainable metrics. In *Proceedings of the 4th Workshop on Evaluation and Comparison for NLP systems*.

Qingyu Lu, Liang Ding, Liping Xie, Kanjian Zhang, Derek F Wong, and Dacheng Tao. 2022. Toward human-like evaluation for natural language generation with error analysis. *arXiv preprint arXiv:2212.10179*.

Qingyu Lu, Baopu Qiu, Liang Ding, Liping Xie, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint arXiv:2303.13809*.

Pankaj Mathur. 2023. orca_mini_v3_7b: An explain tuned llama2-7b model. https://https://huggingface.co/psmathur/orca_mini_v3_7b,.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, 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 of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Nils Reimers 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

A Different Prompts

Prompt 1:
Score the following translation from {source_lang} to {target_lang} with respect to the source sentence on a continuous scale from 0 to 100, where a score of zero means "no meaning preserved" and score of one hundred means "perfect meaning and grammar".
{source_lang} source: "{source}"
{target_lang} translation: "{target}"
Score (0-100): **{score}**

Prompt 2:
Score the following translation from {source_lang} to {target_lang} with respect to the source sentence on a continuous scale from -100 to 0, where a score of negative one hundred means "no meaning preserved" and score of zero means "perfect meaning and grammar".
{source_lang} source: "{source}"
{target_lang} translation: "{target}"
Score (0-100): **{score}**

Prompt 3:
Based on the given source, identify the major and minor errors in this translation. Note that Major errors refer to actual translation or grammatical errors, and Minor errors refer to smaller imperfections, and purely subjective opinions about the translation. Then count the number of major and minor errors and compute the final score for this translation. Deduct 5 points for each major error. Deduct 1 point for each minor error. If the translation has no errors, its score will be 0.
{source_lang} source: "{source}"
{target_lang} translation: "{target}"
Score (0-100): **{score}**

Prompt 4:
Score the following {target_lang} sentence on a continuous scale from 0 to 100, where a score of zero means "grammatically incorrect and bad-written" and score of one hundred means "grammatically correct and well-written".
{target_lang} sentence: "{target}"
Score (0-100): **{score}**

Figure 3: Different prompts used in our generation-based method.

Translate the following {source_lang} sentence into {target_lang}.
{source_lang} source: {source}
{target_lang} translation: {target}

Translate {source} into {target_lang}: {target}
Please translate {source} into {target_lang}: {target}
Help me to translate {source} into {target_lang}: {target}
Translate {source} from {source_lang} into {target_lang}: {target}
Please translate {source} from {source_lang} into {target_lang}: {target}
Help me to translate {source} from {source_lang} into {target_lang}: {target}
{source_lang}: {source}; {target_lang}: {target}
{source_lang} source: {source}; {target_lang} translation: {target}
The {target_lang} translation of {source_lang} is: {source} {target}

Figure 4: Different prompts used in our probability-based method.