

Exploring Automatic Evaluation Methods based on a Decoder-based LLM for Text Generation

Tomohito Kasahara, Daisuke Kawahara

Waseda University

{tomo_k@ruri., dkw@}waseda.jp

Abstract

Automatic evaluation of text generation is essential for improving the accuracy of generation tasks. In light of the current trend towards increasingly larger decoder-based language models, we investigate automatic evaluation methods based on such models for text generation. This paper compares various methods, including tuning with encoder-based models and large language models under equal conditions, on two different tasks, machine translation evaluation and semantic textual similarity, in two languages, Japanese and English. Experimental results show that compared to the tuned encoder-based models, the tuned decoder-based models perform poorly. The analysis of the causes for this suggests that the decoder-based models focus on surface word sequences and do not capture meaning. It is also revealed that in-context learning of very large decoder-based models such as ChatGPT makes it difficult to identify fine-grained semantic differences.

1 Introduction

Neural network-based text generation models are used in various natural language processing tasks, including machine translation, dialogue systems, and text summarization. However, the outputs from these models are open-ended, and there is no single correct answer, making the evaluation of generations difficult. Manual evaluation is often used due to its high accuracy but incurs significant temporal and financial costs. Therefore, automatic evaluation is essential for the rapid development of text generation models.

Automatic evaluation methods for text generation, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), have been based mainly on surface word overlaps between the generated text and the reference text. In recent years, with the development of self-supervised models such as BERT (Devlin et al., 2019) and BART (Lewis

et al., 2020), more accurate automatic evaluation methods have been proposed. For example, BERTScore (Zhang et al., 2020) uses word embeddings obtained by these models. Such methods can be classified along two axes: whether the model used is an encoder-based, decoder-based, or encoder-decoder-based architecture of Transformer (Vaswani et al., 2017), and whether tuning is performed. While encoder-based methods with tuning are reported to be highly accurate (Rei et al., 2020), in-context learning without tuning is the mainstream in decoder-based methods.

In recent years, self-supervised decoder-based models have become larger and larger, as seen in GPT-4 (OpenAI, 2023), Megatron-Turing (Smith et al., 2022), and PaLM (Chowdhery et al., 2022). These decoder-based self-supervised large language models are referred to as LLMs in this paper. However, encoder-based models have remained relatively smaller than decoder-based ones.

Based on the above situation, this paper compares various methods, including tuning with encoder-based models and LLMs under equal conditions, on two different tasks, machine translation evaluation and semantic textual similarity (STS), in two languages, Japanese and English. The results revealed the following three observations.

1. When a decoder-based model is tuned, the accuracy is proportional to the model size up to a certain model size, but it reaches a ceiling.
2. Compared to tuned encoder-based models, tuned decoder-based models perform poorly.
3. In-context learning of very large decoder-based models such as ChatGPT¹ makes it difficult to identify fine-grained semantic differences.

The analysis of the causes for the poor performance of the tuned decoder-based models suggests

¹<https://openai.com/chatgpt>

that they focus on surface word sequences and do not capture meaning. Note that our study focuses on evaluation methods under the assumption that reference text is available.

2 Related Work

Automatic evaluation of text generation mainly requires the text generated by a model and the reference text. The classic automatic evaluation metrics, such as BLEU, ROUGE, METEOR (Banerjee and Lavie, 2005), and CIDEr (Vedantam et al., 2015), are based on the n-gram overlap between these two texts. The biggest disadvantage of these metrics is that they do not score well even when synonyms are included, as the n-grams must match exactly for a higher score. TER (Snover et al., 2006) and others that base their evaluation on edit distance have similar drawbacks. METEOR aims to overcome this drawback by using a synonym dictionary, but it is unable to perform context-sensitive synonym evaluation.

Using embeddings derived from self-supervised models, synonyms can be judged to be similar based on their context. BERTScore (Zhang et al., 2020) is a method that embeds the generated text and the reference text respectively by an encoder-based model and calculates a score based on their similarity. BARTScore (Yuan et al., 2021) and T5Score (Qin et al., 2022) input the source text to the encoder and the target text to the decoder, and calculate a score based on the generation probability of the target text. GPTScore (Fu et al., 2023) calculates a score based on the generation probability of the target text by applying in-context learning (Brown et al., 2020) to an LLM. G-Eval (Liu et al., 2023) proposes a method to have an LLM generate scores directly. In addition, Chen et al. (2023) show that directly generated scores are more accurate than generation probability-based ones when using LLMs.

Other evaluation methods increase accuracy by fine-tuning a self-supervised model using datasets consisting of text pairs and their similarity labels. Models trained on translation evaluation datasets include BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020), while models trained on STS datasets include Sentence-BERT (Reimers and Gurevych, 2019). There are also methods such as SimCSE (Gao et al., 2021) that learn sentence embeddings by contrastive learning on natural language inference datasets and use them to

calculate text pair similarity. Most of these self-supervised methods use encoder-based models. InstructScore (Xu et al., 2023) is a method of fine-tuning LLaMA (Touvron et al., 2023). However, Xu et al. (2023)’s experiments did not involve tuned LLMs on the target datasets and did not compare them to encoder-based models under equal conditions. In this study, we compare LLMs, which do not have bidirectional attention but larger model size, with encoder-based models, which have bidirectional attention but smaller model size, by tuning them under equal conditions.

3 Experimental Setup

We compare various methods for text generation evaluation, including tuned encoder-based models and LLMs on equal conditions, on two different tasks, machine translation evaluation and STS, in two languages, Japanese and English.

3.1 Datasets

3.1.1 Datasets in English

For the experiments in English, we use WMT20 (Mathur et al., 2020) and WMT21 (Fritag et al., 2021) as the translation evaluation datasets, and STS-B (Cer et al., 2017) and SICK (Marelli et al., 2014) as the datasets for STS. WMT20 and WMT21 include human-translated texts, machine-translated texts, and their evaluation labels of Direct Assessment (DA) and Multidimensional Quality Metrics (MQM). In our experiments, we adopted the MQM labels that were evaluated by experts and native speakers. Since only the Chinese-to-English translation task is labeled with MQM, we use its datasets (WMT20 MQM and WMT21 MQM). STS and SICK consist of sentence pairs and their similarity labels. Note that for WMT20 and WMT21, the datasets were not pre-separated into train, valid, and test, and we randomly split these datasets with a ratio of 8:1:1.

3.1.2 Datasets in Japanese

The datasets used in the experiments in Japanese are the WMT20 English to Japanese translation task (WMT20 en-ja) and JSTS included in the Japanese General Language Understanding Evaluation (JGLUE) (Kurihara et al., 2022) benchmark. The WMT20 dataset includes human-translated texts, machine-translated texts, and their evaluation labels (Direct Assessment). JSTS is an STS dataset for Japanese, consisting of sentence pairs and their

Method	Model	Architecture	Size	WMT20	WMT21	STS-B	SICK
No-Tuning Methods							
BLEU	-	-	-	0.109	0.120	0.244	0.354
Edit Distance	-	-	-	0.345	0.340	0.089	0.278
BERTScore	RoBERTa-large	Encoder	355M	0.306	0.294	0.405	0.455
BARTScore CNN+Para	BART-large	Enc-Dec	406M	0.225	0.219	0.475	0.505
OpenAI Embeddings	text-embedding-ada-002	Encoder	?	0.184	0.181	0.655	0.627
ChatGPT Zero-Shot	gpt-3.5-turbo	Decoder	?	0.113	0.097	0.669	0.622
ChatGPT Few-Shot	gpt-3.5-turbo	Decoder	?	0.175	0.136	0.618	0.656
Tuning Methods (Not Target Dataset)							
BLEURT-20	RemBERT	Encoder	576M	0.345	0.323	0.620	0.574
InstructScore	LLaMA	Decoder	6.7B	0.439	0.345	0.471	0.526
Tuning Methods (Target Dataset)							
COMET (WMT21 MQM)	XLm-RoBERTa-large	Encoder	560M	0.506	0.362	-	-
RoBERTa Fine-Tuning	RoBERTa-large	Encoder	355M	0.699	0.391	0.737	0.658
			111M	0.589	0.362	0.540	0.425
			256M	0.634	0.378	0.585	0.462
			590M	0.654	0.371	0.616	0.486
LLM LoRA-Tuning	Cerebras-GPT	Decoder	1.3B	0.663	0.383	0.625	0.483
			2.7B	0.671	0.377	0.661	0.512
			6.7B	0.665	0.370	0.681	0.530

Table 1: Kendall’s correlation coefficients between the predictions by the automatic evaluation metrics and the labels in the experiments in English.

similarity labels. Note that WMT20 en-ja was randomly split at a ratio of train:valid:test=8:1:1 as in the English datasets.

3.2 Tuning of LLMs

For the method by LLM tuning, we performed LoRA-tuning of LLMs using datasets of text pairs and their evaluation or similarity labels. We chose LoRA-tuning because it can achieve competitive accuracy with fine-tuning at a lower cost (Hu et al., 2021).

3.2.1 Architecture and Input-Output Relationships

The architecture and input-output relationship of the LLM’s tuning are shown in Figure 1. Given a text pair as an input to the model, their similarity value is returned as an output. The following procedure is used to calculate the similarity.

1. Feed each text of a text pair into an LLM.
2. Obtain the embedding corresponding to the token at the end of each text (the preceding token of the EOS token).
3. Calculate the cosine similarity between the two embeddings.
4. Pass the cosine similarity to a 1-layer FNN and regard its output as the similarity of the text pair.

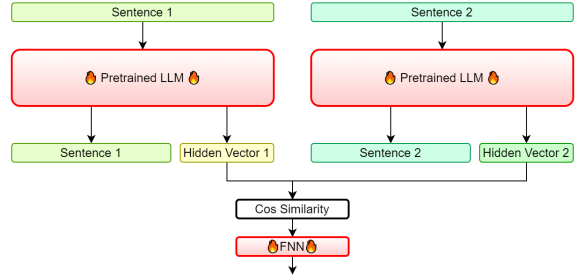


Figure 1: The architecture and input-output overview of the LLM’s tuning.

The FNN layer is used to convert the cosine similarity values into a label distribution of the dataset. Based on the results of our preliminary experiments, we decided to use the embedding of the token at the end of a text instead of the special EOS token.

3.2.2 Training Method

The gold labels (similarity values) in the dataset are normalized between 0 and 1 in advance. We calculate the similarity of a text pair using the procedure described in Section 3.2.1. Next, only the parameters newly added to the model (including the parameters of the FNN) are updated based on the mean squared error between the predictions and the gold labels. Furthermore, the initial values of the FNN are set to 1 for weight and 0 for bias. We employ LoRA-tuning as the tuning method of the LLM for its high performance.

For experiments in English, we use the Cerebras-

Method	Model	Architecture	Size	WMT20	JSTS
No-Tuning Methods					
BLEU	-	-	-	0.226	0.353
Edit Distance	-	-	-	0.242	0.321
BERTScore	Waseda RoBERTa-large	Encoder	337M	0.319	0.558
OpenAI Embeddings	text-embedding-ada-002	Encoder	?	0.237	0.611
ChatGPT Zero-Shot	gpt-3.5-turbo	Decoder	?	0.187	0.709
ChatGPT Few-Shot	gpt-3.5-turbo	Decoder	?	0.205	0.690
Tuning Methods (Not Target Dataset)					
BLEURT-20	RemBERT	Encoder	576M	0.315	0.569
Tuning Methods (Target Dataset)					
RoBERTa Fine-Tuning	Waseda RoBERTa-large	Encoder	337M	0.396	0.729
			37M	0.342	0.600
			110M	0.378	0.644
LLM LoRA-Tuning	Rinna-gpt	Decoder	336M	0.396	0.677
			1.3B	0.370	0.659
			3.6B	0.380	0.687

Table 2: Kendall’s correlation coefficients between the predictions by the automatic evaluation metrics and the labels in the experiments in Japanese.

GPT models² with parameter sizes ranging from 111M to 6.7B. These models are tuned on WMT20 MQM for the translation evaluation task and on STS-B for the STS tasks, respectively. In other words, the models trained with WMT20 MQM are evaluated on WMT20 MQM and WMT21 MQM, and the models trained with STS-B are evaluated on STS-B and SICK.

For experiments in Japanese, we use the GPT-2 and GPT-NeoX models developed by rinna³, ranging from the 37M model to the 3.6B model. We trained models on each of the two datasets in Section 3.1.2.

3.3 Baselines

For comparison, we adopt the following baselines: BLEU, character edit distance, fine-tuned RoBERTa-large (Liu et al., 2019), BERTScore⁴, BARTScore⁵, OpenAI Embeddings (Neelakantan et al., 2022), in-context learning of ChatGPT (gpt-3.5-turbo), BLEURT⁶, COMET⁷ and InstructScore⁸. For fine-tuned RoBERTa, as described in Section 3.2.2, we trained models on WMT20 MQM and STS-B for the English experiments and on the two datasets shown in Section 3.1.2 for the Japanese experiments, respectively. For BERTScore, the training data is used to select the best output layer to obtain the em-

beddings. For OpenAI Embeddings, the scores are the cosine similarity of the obtained embeddings. The prompt used in ChatGPT’s in-context learning is shown in Appendix A. We also had a preliminary experiment with in-context learning of Cerebras-GPT as well as ChatGPT, but were unable to generate scores successfully. It is assumed that the model size of few billion is too small for in-context learning. We do not tune BLEURT, but instead use BLEURT-20 (Pu et al., 2021), which is trained in multiple languages. For COMET, we use the model trained on WMT21 MQM. We do not apply COMET to the STS datasets because COMET is a metric for automatic translation evaluation and requires three inputs: pre-translated text, human-translated text, and machine-translated text. Our hyperparameters for training are shown in Appendix B.

Note that BARTScore, COMET, and InstructScore, only support English and hence are not used for experiments in Japanese.

4 Experimental Results and Analysis

4.1 Main Results

Kendall’s correlation coefficients between the predictions by the automatic evaluation metrics and the gold labels in English and Japanese are shown in Tables 1 and 2, respectively. For all datasets in both languages, RoBERTa-large with fine-tuning achieved the highest accuracy. For LoRA-tuned LLMs, there is a tendency for the accuracy to be proportional to the model size up to a certain model size, but it reaches a ceiling. Also, even models with overwhelmingly larger parameter sizes than

²<https://huggingface.co/cerebras>

³<https://huggingface.co/rinna>

⁴https://github.com/Tiiiger/bert_score

⁵<https://github.com/neulab/BARTScore>

⁶<https://github.com/google-research/bleurt>

⁷<https://unbabel.github.io/COMET>

⁸<https://github.com/xul998hz/SESscore3>

Model	Size	BLEU				Edit Distance			
		WMT20	WMT21	STS-B	SICK	WMT20	WMT21	STS-B	SICK
RoBERTa-large	355M	0.126	0.127	0.237	0.363	0.394	0.511	0.046	0.262
Cerebras-GPT	111M	0.213	0.212	0.281	0.553	0.491	0.612	0.077	0.491
	256M	0.192	0.216	0.292	0.553	0.455	0.615	0.081	0.486
	590M	0.187	0.211	0.268	0.559	0.432	0.583	0.087	0.487
	1.3B	0.175	0.225	0.277	0.545	0.425	0.574	0.096	0.483
	2.7B	0.178	0.231	0.263	0.549	0.428	0.567	0.058	0.478
	6.7B	0.181	0.205	0.259	0.552	0.441	0.522	0.068	0.472

Table 3: Kendall’s correlations between the metrics based on superficial word sequences and the predictions by models with tuning in the experiments in English.

RoBERTa-large showed low accuracy. For ChatGPT’s in-context learning, the accuracy on the STS datasets was comparable to that of the tuning-based methods, but its accuracy on the translation evaluation datasets was low. Note that most of the p-values were very close to 0.

4.2 Analysis of Why Tuned LLMs are Inferior

From Tables 1 and 2, we observe that LoRA-tuned LLMs, which have by far a larger number of parameters than RoBERTa-large, are inferior in terms of performance. We analyze the causes of this from the experimental results in English.

The most significant difference between the two models is that RoBERTa, an encoder-based model, has bidirectional attention, while an LLM has unidirectional attention. Here, we hypothesized that unidirectional attention focuses more on surface word sequences as opposed to bidirectional attention. To confirm this hypothesis, we calculated the correlations of the predictions of RoBERTa and LLMs to BLEU and character edit distance, which are the metrics based on superficial word sequences. The results are shown in Table 3. As hypothesized, the results show that the correlations to both BLEU and edit distance are stronger for LLMs than the encoder-based model. The fact that the correlation decreases as the model size increases in LLMs suggests that the larger the model size, the better the prediction is able to capture not only the surface word sequences but also the meaning of the text. However, even with a model size of 6.7B, the LLM is still not as accurate as RoBERTa.

4.3 Analysis of the Inability of ChatGPT’s In-context Learning

While ChatGPT’s in-context learning showed high accuracy on the STS datasets, it did not perform well on the translation evaluation datasets. We analyze the causes of this from the experimental results in English.

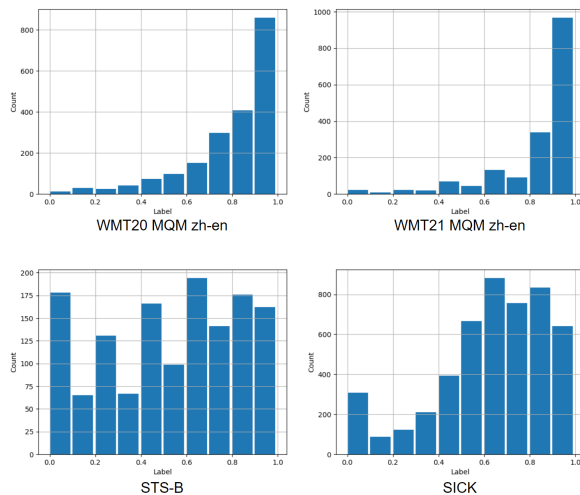


Figure 2: Label distribution of the test datasets used in the English experiments.

In our experiments, the prompts were created to score on a scale of 0 to 100. However, in the output scores, there were many cases where the last digit was 0 or 5 in both zero-shot and few-shot settings. Also, as shown in Figure 2, the label distributions of the translation evaluation datasets are skewed between 0.9 and 1.0, compared to the STS datasets, which have gently sloping distributions. Therefore, most of the predictions in the translation evaluation datasets are 95, etc., and this is thought to have caused the accuracy drop. Thus, it is clear that ChatGPT’s in-context learning has difficulty in identifying fine-grained semantic differences.

5 Conclusion

In this paper, we compared various automatic evaluation methods for text generation in two languages, Japanese and English. We showed that fine-tuned encoder-based models are the strongest when training data is available, and in-context learning of ChatGPT is equally accurate when the variance of scores is large. Our analysis also revealed that tuned LLMs are less accurate than tuned encoder-

based models because of their focus on surface word sequences.

Limitations

Our experiments assume the presence of a training dataset. If no dataset for training exists, refer to the results without the **Tuning Method (Target Dataset)** to compare the metrics in Tables 1 and 2.

Acknowledgements

This work was supported by a joint research grant from LINE Corporation.

References

- Satanjeev Banerjee and Alon Lavie. 2005. **METEOR: An automatic metric for MT evaluation with improved correlation with human judgments**. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- 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. *NeurIPS2020*, 33:1877–1901.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. **SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation**. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *arXiv preprint arXiv:2304.00723*.
- Aakanksha Chowdhery, Sharan Narang, and Jacob Devlin et al. 2022. **Palm: Scaling language modeling with pathways**. arXiv. Abs/2204.02311.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. **Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain**. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GpScore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. **SimCSE: Simple contrastive learning of sentence embeddings**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv. Abs/2106.09685.
- Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. **JGLUE: Japanese general language understanding evaluation**. In *LREC2022*, pages 2957–2966, Marseille, France. European Language Resources Association.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. **ROUGE: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. GpEval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv. Abs/1907.11692.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. **A SICK cure for the evaluation of compositional distributional semantic models**. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

- Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. [Results of the WMT20 metrics shared task](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.
- Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.
- 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. [Learning compact metrics for MT](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2022. T5score: Discriminative fine-tuning of generative evaluation metrics. *arXiv preprint arXiv:2212.05726*.
- Ricardo Rei, 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)*, pages 2685–2702, Online. Association for Computational Linguistics.
- 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, Hong Kong, China. Association for Computational Linguistics.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deep-speed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv. Abs/2201.11990*.
- Matthew Snover, Bonnie Dorr, Rich 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: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NeurIPS2017*.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback. *arXiv preprint arXiv:2305.14282*.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *NeurIPS2021*, 34:27263–27277.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *ICLR2020*.

A Prompt Used in Experiments with ChatGPT

The following text is an example of the prompt used in our experiments with ChatGPT, which was created by referring to the prompt used in [Chen et al. \(2023\)](#)'s experiments.

Score the following text pair on a continual scale from 0 (worst) to 100 (best), where score of 0 means "the meaning of the text pair is completely different" and score of 100 means "the meaning of the text pair is completely identical".

Text 1: There is no boy playing outdoors and there is no man smiling

Text 2: A group of kids is playing in a yard and an old man is standing in the background

Score:

B Hyperparameters

The hyperparameters that we used for training models in our experiments are shown in Table 4. Note that the GPU used in our experiments is the NVIDIA A100 SXM4 GPU with a GPU memory size of 40 GB.

Hyperparameters	RoBERTa	LLM
	Fine-Tuning	LoRA-Tuning
Learning Rate	2e-5	2e-4, 1e-4, 5e-5, 1e-5
Epoch Num	10	10
LoRA Dim	-	4
LoRA Alpha	-	32
LoRA Dropout	-	0.1

Table 4: Hyperparameters for training in the experiments.