Optimizing Machine Translation through Paraphrasing Ranking

Anonymous ACL submission

Abstract

This paper proposes a novel approach for optimizing the performance of a machine translation system. By paraphrasing an input into multiple different phrases, that maintain the semantic meaning, and ranking them using only source-side information, we show that performance can be significantly improved. Experiments on the IWSLT En-De and En-Nl datasets show that the family of Flan-T5 models can be improved by several COMET points, a notable gain in performance. Furthermore, this can be combined with traditional output-side rankers on n-best list outputs to obtain further gains.

1 Introduction

Transformer-based autoregressive systems have achieved state-of-the-art performance in many sequence tasks (Vaswani et al., 2017) including (multilingual) Machine Translation (MT) (Xue et al., 2021; Costa-jussà et al., 2022), Text Summarization & Generation (Chung et al., 2022) and Speech Recognition (Chiu et al., 2018; Gulati et al., 2020; Radford et al., 2022). By training these systems using the next-token prediction of a single reference sequence, impressive performance can be obtained. However, two issues plague such approaches. Firstly, in tasks such as MT and Text Summarization, there exist several plausible answers for every input but the model is trained to allocate all probability mass to a single reference (Zhang et al., 2020; Liu et al., 2022). Secondly, such systems suffer from exposure bias; the model is only trained to predict the next token conditioned on a reference back-history, but not its own generations (Williams and Zipser, 1989; Bengio et al., 2015; Lamb et al., 2016; Gu et al., 2019; Wiseman et al., 2016; Kim and Rush, 2016).

A direct consequence of the above-mentioned issues is the uncalibrated confidence scores that are produced by such systems. Prior work has found that although good performance can be achieved, the confidence scores across a set of hypotheses (generated through beam search) correlate weakly with the quality of the hypotheses (Fathullah et al., 2023; Zhao et al., 2023). One family of approaches that attempts to solve this modify the training approach to incorporate several targets and allocate a probability mass that correlates with the quality of the target (Liu et al., 2022; Zhao et al., 2023). However, while such approaches improve the correlation between confidence and quality, they require modifying the parameters of the system. When operating foundation models that are either highly expensive to train, or are hidden behind application program interfaces (APIs), such approaches become less practical (Raffel et al., 2020; Brown et al., 2020; Touvron et al., 2023a,b; Achiam et al., 2023; Anil et al., 2023).

We take an alternative approach based on ranking models (Shen et al., 2004). The main aim of a ranker system is to select the best hypothesis in a decoding set generated by a model, according to some predetermined performance metric. This is traditionally achieved by training such a system, conditioned on the input and hypothesis, to output scores that are directly correlated with the quality of the hypothesis (Shen et al., 2004; Och et al., 2004; Salazar et al., 2020; Lee et al., 2021). While prior work, to the best of our knowledge, aims to rank the outputs of a system, we propose paraphrasing inputs and choosing the one that would lead to the best hypothesis. By maintaining the semantic meaning of an input sequence, a paraphrased version could, for example, trigger an MT system to generate better translations. Experiments on IWSLT translation datasets (Cettolo et al., 2017) show that paraphrasing indeed has the potential to improve translation performance for large foundation models (such as Flan-T5 (Raffel et al., 2020)) and can be further improved by combining it with output ranking to obtain better results.

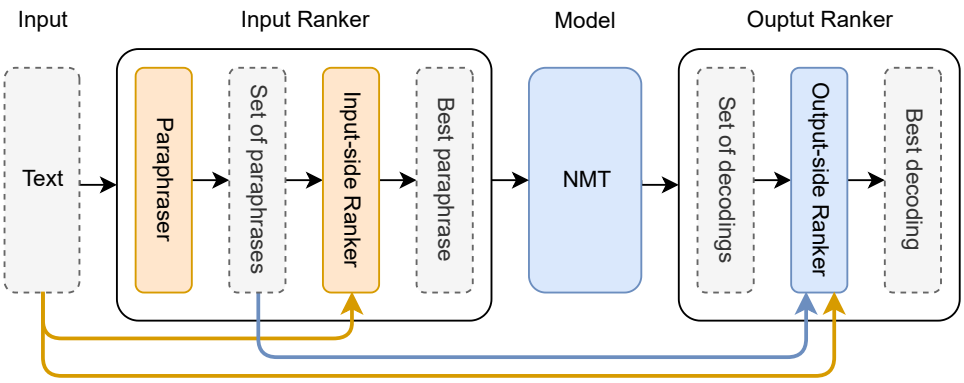

Figure 1: Ranking setup. On the input side, a ranker picks the paraphrase that should give the best decoding. Optionally, the NMT model could also produce a set of decodings which an output-side ranker can rank. Note that the output ranker explicitly incorporates the paraphrased information.

2 Background

In this current paradigm of large foundation models that often are hidden behind APIs, approaches that aim to modify the parameters of the system are either expensive or not possible. Instead, methods that aim to modify the input or select the best output have become more practical. In the field of Machine Translation, there has been a range of work on improving the quality of translations through the use of auxiliary networks that aim to rank a set of hypotheses (Shen et al., 2004; Och et al., 2004; Salazar et al., 2020; Lee et al., 2021). The work of Salazar et al. (2020) used masked language models (MLMs) (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020) to rank candidates. Their approach was centred around masking one token at a time and extracting the log-likelihood to obtain an overall confidence score. While this approach is expensive and is not directly tailored to ranking hypotheses, they showed it was possible to obtain better performance using off-the-shelf MLMs. Furthermore, Lee et al. (2021) proposed fine-tuning MLMs to directly produce scores correlated with the metric of interest. By conditioning an MLM (with parameters $\boldsymbol{\theta}$) on the source input $\boldsymbol{x}$ and a hypothesis $\boldsymbol{u}_i \in \mathcal{U}$ in a candidate list, it is tasked with producing scores $o(\boldsymbol{u}_i|\boldsymbol{x}, \boldsymbol{\theta})$ such that the resulting distribution:

$$\mathrm{p}(\boldsymbol{u}_i|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp\left(o(\boldsymbol{u}_i|\boldsymbol{x}, \boldsymbol{\theta})\right)}{\sum_{\boldsymbol{u}\in\mathcal{U}} \exp\left(o(\boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\theta})\right)}$$

matches the target distribution derived from performance metric π such as COMET:

$$\mathrm{p}(\boldsymbol{u}_i|\boldsymbol{x}) = \frac{\exp\left(\pi(\boldsymbol{u}_i, \boldsymbol{r}|\boldsymbol{x})/T\right)}{\sum_{\boldsymbol{u}\in\mathcal{U}} \exp\left(\pi(\boldsymbol{u}, \boldsymbol{r}|\boldsymbol{x})/T\right)} \quad (1)$$

where $\boldsymbol{r}$ is the reference for some input $\boldsymbol{x}$. While effective, this approach relies on crafting a target distribution with some predetermined temperature T which could affect training and performance. Finally, the work of Fathullah et al. (2023) showed it was possible to fine-tune MLMs to predict the performance of a translation system using only source information. This was achieved by taking a batch of inputs $\boldsymbol{x} \in \mathcal{X}$, their corresponding decodings $\boldsymbol{u}$ and references $\boldsymbol{r}$ and trained the system $o(\boldsymbol{x}|\boldsymbol{\theta})$ to achieve a high correlation with the metric of interest $\pi(\boldsymbol{u}, \boldsymbol{r}|\boldsymbol{x})$. Note that such a system is used to compare different instances while traditional ranking systems compare different hypotheses for the same instance.

3 Paraphrasing Inputs

At the core of our proposal, we ask the following question: *Can we improve the performance of a translation system by modifying the input?* Our answer to this requires two components: (1) a mechanism for modifying inputs, in our case a paraphrasing system which can modify a sentence while maintaining its semantic meaning. (2) a model for evaluating whether a new paraphrase would trigger better translations or not. If both components can be obtained, then it would be possible to modify an input, evaluate whether it is a high-quality modification, and pass it on to the MT system for translation and achieve better results, see Figure 1. The first component is trivial, efficient paraphrasing systems (Vladimir Vorobev, 2023) exist and can be used off-the-shelf from HuggingFace (Wolf et al., 2019) for our task. The second component can be seen as an input ranking system. Taking

Table 1: COMET performance of various **trained** and **oracle** rankers on the `IWSLT-2017 En-De` test set. The first block shows the use of an input-side ranker with greedy decoding of the NMT system. The second block shows an additional use of an output-side ranker on the beam output. The `Fraction Improved` column refers to the fraction of the dataset that was improved through input paraphrasing and ranking.

Model	Greedy Decoding			Beam Search Decoding			
	Base.	Input	Fraction Improved	Base.	Input	Ouptut	Input & Output
Flan-T5 Small	46.8	51.1 (+4.3)	52.9%	53.8	55.1	57.3	58.3 (+4.5)
– oracle		58.0 (+11.2)	78.5%		61.0	62.3	68.0 (+14.2)
Flan-T5 Base	57.7	61.4 (+3.7)	47.0%	62.1	64.7	65.6	68.3 (+6.2)
– oracle		68.4 (+10.7)	75.2%		68.9	69.9	74.7 (+12.6)
Flan-T5 Large	65.0	67.9 (+2.7)	37.2%	67.7	70.9	71.7	74.1 (+6.4)
– oracle		74.5 (+9.5)	69.6%		75.2	75.7	79.8 (+12.1)
Flan-T5 XL	69.3	71.7 (+2.4)	26.8%	69.6	72.0	73.5	75.6 (+6.0)
– oracle		77.7 (+8.4)	65.5%		77.1	77.9	81.6 (+12.0)
Flan-T5 XXL	72.4	74.2 (+1.8)	24.8%	74.3	76.0	77.7	78.8 (+4.5)
– oracle		79.8 (+7.4)	62.3%		80.8	80.6	83.7 (+9.4)

only input-side information its task is to discriminate between different paraphrases and locate the one that would induce the best translation. From this perspective, our approach draws inspiration from Fathullah et al. (2023); Lee et al. (2021) on metric estimation using input-side information and discriminative ranking.

Let $\boldsymbol{x}$ be some input and let $\tilde{\boldsymbol{x}}_i \in \tilde{\mathcal{X}}$ be a set of paraphrases that include the original input. For each paraphrase $\tilde{\boldsymbol{x}}_i$ the MT system generates a decoding $\tilde{\boldsymbol{u}}_i$ with a performance metric $\pi_i = \pi(\tilde{\boldsymbol{u}}_i, \boldsymbol{r}|\boldsymbol{x})$, where $\boldsymbol{r}$ is the reference translation for $\boldsymbol{x}$. The task of the input-ranker is to take the input-side information and predict a score $o_i = o(\boldsymbol{x}, \tilde{\boldsymbol{x}}_i|\boldsymbol{\theta})$ that is correlated with the metric.

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{\sum_i (o_i - \mu_o)(\pi_i - \mu_\pi)}{\sqrt{\sum_i (o_i - \mu_o)^2}\sqrt{\sum_i (\pi_i - \mu_\pi)^2}}$$

To train the input-ranker we task it with optimising the Pearson Correlation (see equation above) across a set of candidates. Note that this loss function requires no hyperparameters and is simpler than capturing the heuristic target distribution in Eq. (1). Furthermore, we use a correlation loss since absolute predictions are not required to discern between candidates.

4 Experimental Evaluation

All experiments will be conducted on the IWSLT-2017 En-De and En-Nl translation datasets using the Flan-T5 family of foundation models and COMET (`Unbabel/wmt22-comet-da`) (Rei et al., 2020) to measure performance. To generate paraphrases we use Vladimir Vorobev (2023) from HuggingFace, with diverse beam search (Vijayakumar et al., 2016) with 8 beams and 8 beam groups to ensure a level of diversity in the paraphrases. All ranking systems will be based on a DeBERTaV3 (He et al., 2023) base backbone with an attention layer and a small multi-layer perceptron on top to predict a scalar score. Furthermore, we modify the discriminative output ranker in Lee et al. (2021), to additionally be conditioned on the paraphrased input. Finally, the NMT system will either produce outputs using greedy search or beam search with 8 beams. Full details on the training setup and hyperparameters are provided in Appendix A.

Table 1 shows the performance of trained and oracle rankers for both input and output-side systems. All of the rankers are lightweight transformer encoder DeBERTaV3 base models with approximately 198 million parameters that are conditioned on the input (and output) information. Focusing on the left-hand block, it isolates the contribution of input rankers when greedy decoding of the NMT system is used: (1) all systems benefit notably by introducing a paraphraser and input-ranker to optimize the performance of the NMT system. Flan-T5 Small was improved by 4.3 COMET points while the largest Flan-T5, the XXL improved by 1.8 points. Despite being small improvements com-

pared to the theoretical gains that can be achieved under an oracle ranker, these still represent notable improvements in system performance. (2) The table also shows that with greedy decoding between 25-50% of samples can be improved under such a scheme while an optimal system could improve 60-80% of samples. (3) Furthermore, we observe that the smaller less robust Flan-T5 models benefit much more input ranking, larger more robust systems already perform well and are harder to improve upon. The second block of the table shows the performance of input and output rankers when separate and combined: (1) both input and output rankers greatly benefit the baseline system, but output rankers are slightly more effective overall. (2) the combination of both rankers shows further gains, i.e. input and output-side rankers complement each other. While the theoretical gains are significantly larger, the combination can still achieve between 4.5-6.4 COMET point gains. For example, the combination of an input and output-side ranker for the Flan-T5 Large obtained a significant improvement of 6.4 COMET points and is on par with the baseline Flan-T5 XXL model. The Flan-T5 XL system gained 6.0 COMET points and outperformed the XXL baseline.

Next, we explore the performance of our proposal on the IWSLT En-Nl dataset. Since the Flan-T5 family was trained on less Dutch (Nl) text (compared to German; De), we naturally expect the performance of these systems to be lower. However, we can utilize that the paraphraser and input-ranker operate on the high-resource side of English to improve performance. Even if the system has poor performance on the low-resource target language, good paraphrases in the high-resource source language could trigger the system to perform much better. We also included the dedicated NLLB-200 Distilled 600M NMT system as a point of reference since this system should be very robust on this task (Costa-jussà et al., 2022). From Table 2 we observe significant practical and theoretical gains for the Flan-T5 family. Compared to the En-De counterpart, we observe larger gains in the 3.1-9.7 COMET point range. While the Flan-T5 Small obtains a smaller gain, possibly due to its too poor performance on this task, the remaining models can benefit greatly from the use of a modified and improved input to perform the translation. Even a dedicated and robust NMT system such as the NLLB-200 can be improved by almost a COMET point, but as expected, there are diminishing returns when attempting to improve better-performing systems. Overall, this shows that one can improve the translation into low-resource languages by exploiting the high-resource source language and choosing a more appropriate input.

Table 2: COMET performance of various input-side rankers on the IWSLT-2017 En-Nl test set.

Model	Greedy Decoding		
	Base.	Input	Fraction Improved
Flan-T5 Small	26.0	29.1 (+3.1)	60.0%
– oracle		33.4 (+7.4)	85.9%
Flan-T5 Base	32.3	38.1 (+5.8)	60.3%
– oracle		44.5 (+13.2)	84.2%
Flan-T5 Large	33.2	42.9 (+9.7)	54.3%
– oracle		49.5 (+16.3)	86.6%
Flan-T5 XL	38.8	46.2 (+7.4)	68.4%
– oracle		51.9 (+13.1)	82.7%
Flan-T5 XXL	49.6	53.9 (+4.3)	52.5%
– oracle		62.2 (+12.6)	80.6%
NLLB-200	84.3	85.1 (+0.8)	19.0%
– oracle		87.2 (+2.9)	37.8%

5 Conclusion

This paper has showcased a novel approach to improving the performance of translation systems through paraphrasing inputs. By generating a list of candidate paraphrases of a certain input, and ranking them, it is possible to trigger a translation system to output higher-quality decodings. Experiments on the IWSLT-2017 En-De and En-Nl translation datasets show that the proposed lightweight input-side rankers can pick better paraphrases and improve the performance of Flan-T5 models by several COMET points, signifying a notable improvement.

Limitations

This work has focused on translation to showcase how generating paraphrases to an input and ranking them can improve performance. While not explored in this work, the performance of the paraphraser (Vladimir Vorobev, 2023) is very important in ultimately determining the performance of the system and future work could focus on improving the paraphrase generation process. Furthermore, there has been a significant shift to the use of large language models (LLMs) to perform a variety of tasks through the use of intelligent prompting. Our approach could potentially be extended to gener-

ate alternative prompts that could make better use of LLMs but in order to achieve this, much more data, more powerful paraphrasers and rankers are required to be able to encompass the tasks that an LLM can perform. Finally, for alternative tasks such as abstractive summarization, directly paraphrasing long-form text inputs could be computationally expensive.

References

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

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Conference on Neural Information Processing Systems*.

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.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuitho Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the iwslt 2017 evaluation campaign. *International Workshop on Spoken Language Translation*.

Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4774–4778. IEEE.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Association for Computational Linguistics*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yassir Fathullah, Puria Radmard, Adian Liusie, and Mark J. F. Gales. 2023. Who needs decoders? efficient estimation of sequence-level attributes. *arXiv, arXiv:2305.05098*.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein transformer. *Conference on Neural Information Processing Systems*.

Anmol Gulati, Chung-Cheng Chiu, James Qin, Jiahui Yu, Niki Parmar, Ruoming Pang, Shibo Wang, Wei Han, Yonghui Wu, Yu Zhang, and Zhengdong Zhang. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Interspeech*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. *Conference on Neural Information Processing Systems*.

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

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, arXiv:1907.11692*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv, arXiv:2212.04356*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *Association for Computational Linguistics*.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Maxim Kuznetsov Vladimir Vorobev. 2023. A paraphrasing model based on chatgpt paraphrases.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

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*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *International Conference on Learning Representations*.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. 2023. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*.

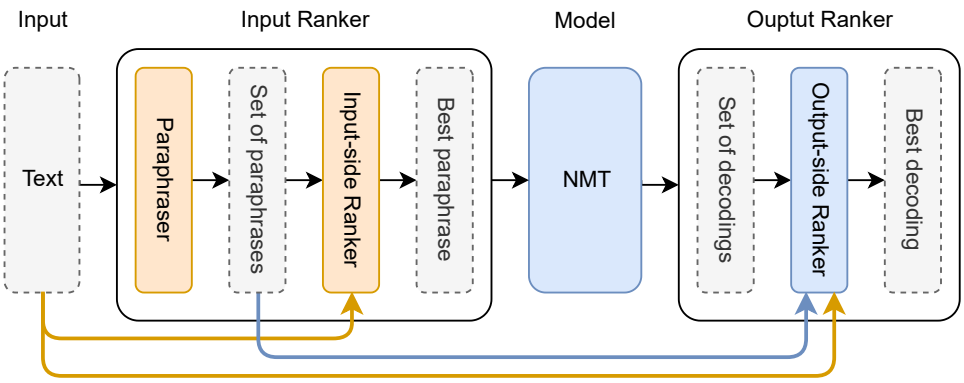

Figure 2: Ranking setup. On the input side, a ranker picks the paraphrase that should give the best decoding. Optionally, the NMT model could also produce a set of decodings which an output-side ranker can rank. Note that the output ranker explicitly incorporates the paraphrased information.

A Setup & Training Details

This section will cover details of experiments. See Figure 2 for a visual setup of the approach. In all experiments, the input and output rankers are composed of two components: (1) MLM back-bone and (2) a small head consisting of an attention layer with a single trainable query in order to pool the encoder output sequence, followed by three linear layers (Fathullah et al., 2023).

Data Generation: To train the input-side rankers, we first took the IWSLT-2017 (Cettolo et al., 2017) dataset and generated 8 paraphrases for each example using diverse beam search with the following parameters:

- `num_beams` $= 8$
- `num_beam_groups` $= 8$
- `repetition_penalty` $= 10.0$
- `diversity_penalty` $= 3.0$
- `no_repeat_ngram_size` $= 2$
- `max_length` $= 128$

through the HuggingFace library (Wolf et al., 2019) with the Humarin system (Vladimir Vorobev, 2023). These parameters were chosen to ensure diversity in the paraphrases. Next, each paraphrase (including the original) was translated using Flan-T5 with greedy decoding and scored using COMET (`Unbabel/wmt22-comet-da`) with the original source input.

Ranker training: As outlined in Section 3, the input-side rankers are trained by taking both the source $\boldsymbol{x}$ and a paraphrase $\tilde{\boldsymbol{x}}_i$ to predict a score $o_i = o(\boldsymbol{x}, \tilde{\boldsymbol{x}}_i|\boldsymbol{\theta})$ that is correlated with the metric of interest $\pi_i = \pi(\tilde{\boldsymbol{u}}_i, \boldsymbol{r}|\boldsymbol{x})$ where the $\tilde{\boldsymbol{u}}_i$ is the decoding corresponding to the paraphrase and $\boldsymbol{r}$ is the reference. The loss function used to train these is the (negative) Pearson Correlation. Similarly, the output ranker is trained by additionally being conditioned on the decoding $o_i = o(\boldsymbol{x}, \tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i|\boldsymbol{\theta})$ which makes it a more powerful ranker but requires the potentially large NMT system to first generate outputs. In our experiments, we found that using the criteria in Lee et al. (2021) to be unstable and defaulted instead to the Pearson Correlation which keeps our experiments consistent across models.

Hyperparameters: All experiments used the same hyperparameters. The DeBERTaV3 base (He et al., 2023) back-bone was not frozen and all rankers were trained using AdamW (Loshchilov and Hutter, 2019) with the following parameters:

- `learning_rate` $= 0.00002$
- `betas` $= (0.9, 0.999)$
- `epsilon` $= 1e-8$
- `weight_decay` $= 0.01$
- `batch_size` $= 4$
- `gradient_accumulations` $= 4$

for only a single epoch. No validation was performed and the final checkpoint was used for evaluation. Each experiment was repeated 3 times using a single NVIDIA A100 80GBs. The training required approximately 2-3 GPU hours per seed.