A LOW-RESOURCE FRAMEWORK FOR DETECTION OF LARGE LANGUAGE MODEL CONTENTS

Linh Le, Shashank Hebbar
College of Computing and Software Engineering
Kennesaw State University
Marietta, GA 30060, USA
lle13@kennesaw.edu, shebbar1620@gmail.com

My Nguyen
Metropolitan College
Boston University
Boston, MA 02215, USA
mytng@bu.edu

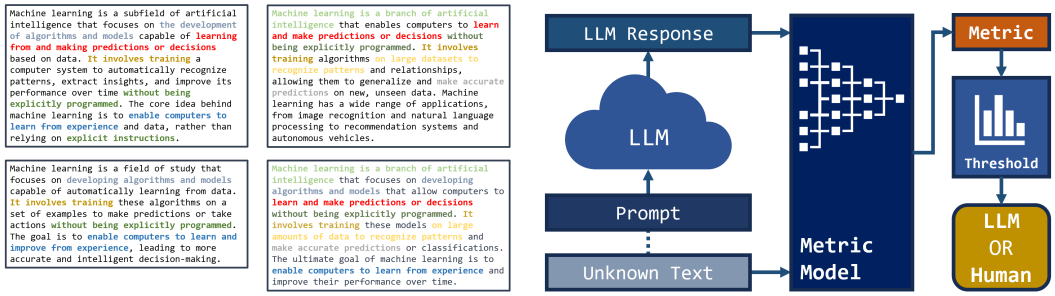
ABSTRACT

Current Large Language Models (LLMs) are able to generate texts that are seemingly indistinguishable from those written by human experts. While offering great opportunities, such technologies also pose new challenges in education, science, information security, and a multitude of other areas. To add up, current approaches in LLM text detection either are computationally expensive or need the LLMs' internal computational states, both of which hinder their public accessibility. To provide better applications for users, especially, in lower-resource settings, this paper presents a new paradigm of metric-based detection for LLM contents that is able to balance among computational costs, accessibility, and performances. Specifically, the detection is performed through utilizing a metric framework to evaluate the similarity between a given text to an equivalent example generated by LLMs and determine the former's origination. Additionally, we develop and publish five datasets totalling over 95,000 prompts and responses from human and GPT-3.5 TURBO or GPT-4 TURBO for benchmarking. In terms of performances, our framework maintains 90-150% F1 scores of a finetuned RoBERTa, while only spends 20-60% of times in training and inference across experiment settings.

1 INTRODUCTION

The advancement in computing technologies has enabled the emergence of large language models (LLMs) (Zhao et al., 2023) packaging up to hundreds of billions of parameters. Examples of recent LLMs are GPT-3 (Brown et al., 2020) at 175 billion parameters, PaML (Chowdhery et al., 2022), 540 billion parameters, and GPT-4, (OpenAI, 2023) 170 trillion parameters. These technologies have brought tremendous potentials to numerous aspects of lives. However, LLMs also come with major challenges. It is increasingly more difficult to determine if texts are written by human or LLMs. This poses a major issue in multiple areas such as education, science, and information security. To add up, detection methods for LLM contents are not widely accessible. A large number of detection methods rely on training or finetuning computationally expensive supervised classifiers (Guerrero & Alsmadi, 2022). Other detection algorithms such as DetectGPT (Mitchell et al., 2023) or watermarking (Kirchenbauer et al., 2023) need access to the LLMs' internal computations, which is not always available to the public and therefore severely hinder their accessibility.

Aiming to assist a wider range of users, especially those who are using lower-resource systems, ***we propose a metric-based approach for LLM content detection that is balanced among computational costs, accessibility, and performances***. Specifically, the detection methods will rely on *comparing a given text to an AI-generated reference from LLMs*. Having the assistance from the generative models, a large part of the computational burdens will be relieved from the detection models. In terms of architectures, the detection framework consists of a pretrained embedding language model and an empirically-designed deep metric network. The metric network is trained to signify the similarity between LLM responses while decreasing that between LLM and human responses. During the decision making phase, the context of a given text is first prompted to a LLM to obtain a LLM-reference. The text and the LLM-equivalence are then fed to a metric framework to obtain their similarity metric. Finally, the metric is compared against a selected threshold to determine the text origination.



(a) GPT-4 TURBO responses for the same context

(b) Decision making framework

Figure 1: Example of repetition in LLM texts and the metric-based detection framework

A metric model can be trained either in pairs or in triplets of instances. Therefore, we further develop five text datasets in the form of context - triplets of responses (one from human and two from LLMs). Using GPT-3.5 TURBO, we obtain $59,945$ entries from the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), $18,813$ from the Stanford Question-Answering Database (SQUAD) (Rajpurkar et al., 2016), $4,419$, Scientific Questions (SciQ) dataset (Johannes Welbl, 2017), and $2,071$, from Wikipedia Scientific Glossary (Wiki) (Wikipedia, 2017). Additionally, we sample $10,290$ prompts and human texts the NQ and SQUAD data and generate LLM texts using GPT-4 TURBO. Experiment studies show that our beset architectures maintain F1 scores mostly in between 0.87 to 0.95 across the tested corpora in both same-corpus and out-of-corpus settings, either with or without paraphrasing. Our framework also achieves much better training and inference time (20 - 60%) than a supervised RoBERTa (Liu et al., 2019). To sum up, our **contributions** are as follows.

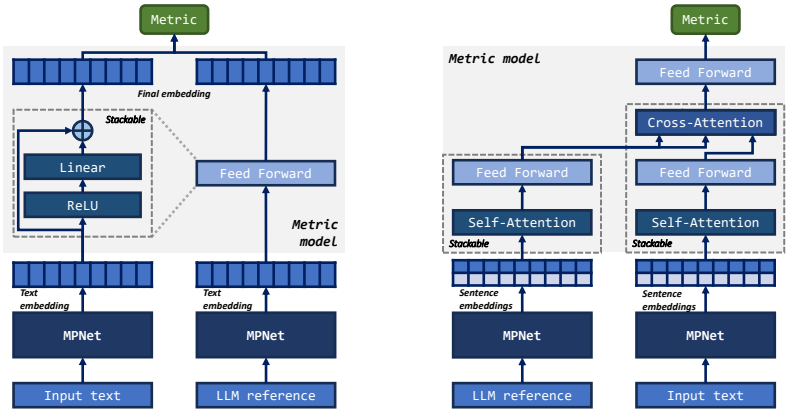
1. A metric-based approach that detects LLM contents that is more light-weighted and does not require access to any LLMs' internal computations. Instead, a given text is compared to an equivalent reference from LLMs. The similarity of the two responses decides if the former was written by LLM or a human.
2. Empirically designed end-to-end deep architectures that transform text data into embedding vectors of which distances between LLM texts are minimized and that between LLM and human-written texts are maximized. The architecture is trained using a same-context sampling strategy to further reduce complexity.
3. Five text datasets totaling over 95,000 instances of contexts and triplet of responses in topics ranging from daily lives to sciences for benchmarking. All datasets will be available for the community after the publication of this paper.

The rest of the paper is organized as follows. Section §2 presents the developed methodologies in details, including the detection algorithm, the sampling strategy, and the framework architecture. Our experiment study is discussed in Section §3. Finally, we conclude our paper in Section §4.

2 METHODOLOGY

Due to the probabilistic models that LLMs use to create their contents, the same context will result in repeated patterns in the generated texts (Welleck et al., 2019; 2020). Key technologies such as *autoregressive generation* (Graves, 2013; Sutskever et al., 2014), *beam search* (Wiseman & Rush, 2016), and *next-token sampling* (Fan et al., 2018; Holtzman et al., 2019) yield outputs based on probabilities of vocabulary tokens being the next ones, all of which are initialized from the starting context. Therefore, same contexts will tend to result in similar pools of tokens for selection which leads to repetitions of patterns. As an illustration, Figure 1(a) shows an example of responses from GPT-3.5 TURBO for the prompt *"briefly explain machine learning"* in four separate sessions. Similar phrases across the responses are bolded and highlighted with the same colors.

With this observation, the detection framework will first learn to optimize similarities among text data. To make decision, a given text is compared to a LLM-generated reference. If the similarity is over a selected threshold, the text is classified as originated from LLM, otherwise, a person. This process is illustrated in Figure 1(b). Next, we discuss the metric learning problem in the context of LLM content detection and the architecture of the complete framework, respectively.



(a) Full-text-level framework (b) Sentence-level framework

Figure 2: Metric neural network architectures

2.1 Metric Learning with Same-Context Sampling

Metric models can be trained in *pairs* or *triplets* of inputs. Training in pairs, the model learns to maximize similarities of texts generated by LLMs and minimize that between human texts and LLM texts through Contrastive Learning (Bengio & Delalleau, 2009). Given a metric model $\mathcal{M}(\cdot)$, the constrastive learning objective is to minimize $\sum_{data}(\mathcal{M}(X_i, X_j) - Y_{ij})^2$ with X_i and X_j being two text instances in training; $Y_{ij} = 0$ if both X_i and X_j are from LLMs, and $Y_{ij} = 1$ otherwise. On the other hand, triplet training (Schroff et al., 2015) utilizes two inputs from LLMs, X_i and X_j, and one from human, X_h. The training objective in this case is to learn a metric that is smaller between two LLM texts, and higher when one input is from human. Mathematically, the model learns through minimizing $\sum_{data} max(\mathcal{M}(X_i, X_j) - \mathcal{M}(X_h, X_j) + \alpha, 0)$ where α is a margin hyperparameter.

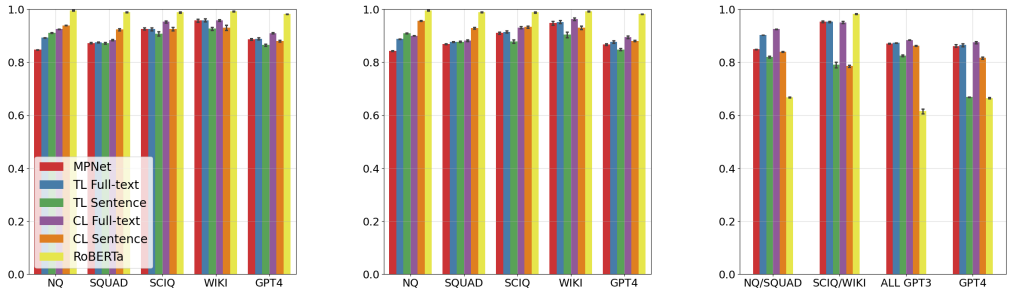
Sampling pairs or triplets randomly, the models have to optimize similarities among texts from different topics which may result in a more complex training problem. This complexity then either requires larger numbers of parameters or makes the models difficult to converge. To simplify the problem, ***pairs or triplets will only come from similar contexts***. Overall, the ultimate goal of metric learning in identifying LLM generated contents is to increase similarity among LLM texts from similar contexts, and decrease that between LLM texts and human texts, under the same condition.

2.2 Model Architecture

The overall framework consists of an embedding model and a metric model. The embedding language model vectorizes raw text data which are then fed to the metric model to output their distance values. Decision making is performed based on the final output distances. To save computational resources as well as utilize knowledge from external domains, we use a pretrained MPNet (Song et al., 2020) which is available in the SentenceTransformer (Reimers & Gurevych, 2019) library. We then develop two metric models, one at the full-text level, and the other, the sentence level.

The first metric model takes two full-text embeddings from MPNet and transforms them to higher-level vector representations. Then, the Euclidean distance of the two vectors is computed and used as the metric of the two original inputs. Architecture-wise, the full-text model consists of stacked residual blocks similar to that in the transformer model (Vaswani et al., 2017), each of which has a Rectified Linear Unit (ReLU) layer and a linear layer. The output of the linear layer is then added with the original block input and normalized to the final block output.

The sentence-level metric network receives an array of MPNet sentence embeddings from each input and is similar to the transformer architecture. Specifically, the model has two towers that take MPNet sentence embeddings from the input text and its LLM reference. Each set of embeddings undergoes a set of self-attention and feed-forward blocks. The outputs from the feed-forward blocks of the two towers are then merged in a cross-attention block. The block of self-attention, feed-forward, and cross-attention, is stackable as needed. Lastly, the output of the final cross-attention block is fed to a feed-forward architecture, flattened, and go through a single-output Sigmoid layer to generate the metric of the two original inputs. An illustration of the two frameworks is in Figure 2.



(a) Same-corpus (b) Paraphrased same-corpus (c) Out of corpus

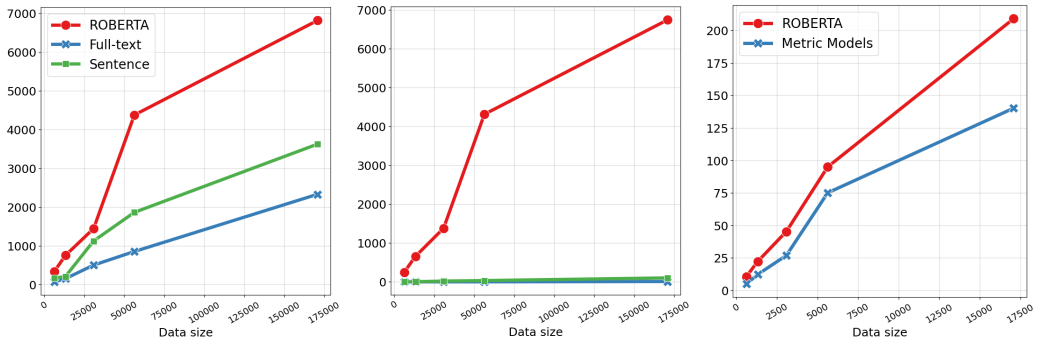
Figure 3: Models' F1 (a)(b)(c) in experiment study

3 Experiment Study

We test our models in three experiment settings: same-corpus, same-corpus with paraphraser, and out-of-corpus. In the two same-corpus settings, the five datasets mentioned earlier are split into 80% training, 10% validation, and 10% testing. In the same-corpus with paraphraser experiment, we apply the Parrot model (Damodaran, 2021), which is a finetuned T5 language model (Raffel et al., 2020), on all inputs. Then, the original version and paraphrased version of the training and validation data are merged, whereas the testing data only comes from the paraphraser. Lastly, in the out-of-corpus setting, we perform four training/testing pairs: NQ/SQUAD, SciQ/Wiki, NQ+SciQ/SQUAD+Wiki (All GPT3), and GPT4-NQ/GPT4-SQUAD (GPT4). In all cases, the ratio of training-validation is $90\% - 10\%$. After finetuning, the final full-text architecture consists of three residual blocks and the sentence framework consists of three attention block and three feed-forward block. For models trained with triplet loss, α is set at 0.25. In all architectures, the number of neurons in all blocks' hidden layers are fixed at 768 which is the dimensionality of embeddings output by the pretrained MPNet. To evaluate our model, we use F1 score. Furthermore, to make prediction, a distance threshold is selected. If a response's distance to its corresponding LLM reference is above the threshold, the response is labeled as human-written, otherwise, it is LLM-generated. We finetune the distance threshold separately each test run by optimize thing F1 in the validation data.

As a baseline, we apply the distance threshold approach on the embedding generated by the standalone pretrained MPNet (denoted *MPNet*). We also attempted to train classifiers on the responses' MPNet embeddings, however, ***these supervised classifiers failed to converge*** even at much higher numbers of parameters compared to the metric models (up to 20 layers - 12 million parameters). While computationally expensive, we include finetuning a large language model, namely, RoBERTa, as another baseline. For illustration, using a T4 graphical unit, ***the full-text model uses*** 8 ***seconds for one epoch*** in the NQ data (about $51,250$ triplets in training and validation), and the sentence model needs about three minutes. In contrast, ***one epoch of finetuning DistilBERT*** (Sanh et al., 2019) on the NQ data split takes ***70 minutes***, and ***RoBERTa*** (Liu et al., 2019), ***125 minutes***. Probabilistic and watermarking approaches are not considered as they need accesses to internal computations of LLMs and do not fit in our target accessibility.

To measure performance, all models are tested in 10 runs. Hyper-parameters are fixed across runs of the same experiment settings. The model F1 are illustrated as bar charts in Figure 3. First, RoBERTa outperforms all models in the same corpus settings, however, the metric models maintain 85-95% of its performance. In the out-of-corpus setting, RoBERTa drops significantly to about 0.67 in three over four experiments. In metric models, the baseline MPNet models perform well in all experiments and achieve F1 of $0.84 - 0.96$. Furthermore, contrastive learning models outperform triplet learning ones in the majority of experiments. In terms of data granularity, sentence-level models seem to obtain very good results but only when the data size is large enough, i.e., in the NQ or SQUAD data . This is explainable as their higher complexity leading to more data required to generalize well, especially to out-of-corpus data. More specifically, the sentence model largely outperforms others in NQ and SQUAD data in same-corpus settings, however, gradually becomes worse in the GPT4, SciQ, and Wiki data. In the out-of-corpus setting, sentence models generalize quite poorly, especially, the triplet learning ones.



(a) Epoch time with embedding (b) Epoch time without embedding (c) Inference time

Figure 4: Comparison of metric models and RoBERTa in time complexity in seconds (c)(d)(e)

Finally, we compare the metric framework to RoBERTa in time complexity as showed in Figure 4 (a)(b)(c). In our experiments, the variation in time of the metric models is negligible, therefore, we generalize them as metric model. The five markers in the plots represent the datasets, i.e., Wiki, SciQ, GPT4, SQUAD, and NQ, in increasing order of data sizes (number of individual texts). For one training epoch, the metric model only needs approximately 20-30% of the time that RoBERTa needs to converge, as in Figure 4 (a). This result, however, also include the embedding time of MPNet before the metric model actually starts training. The time to train a single epoch is significantly lower in the metric model, from milliseconds to below 10 seconds, compared to over two hours that RoBERTa needs, as in Figure 4 (b). This difference will be further exaggerated if RoBERTa needs more epoch to converge, or if the metric model utilizes a faster embedding language model. Finally, in terms of inferences, the difference in time does narrow down, however, the metric model is still faster at 50-60% time needed compared to RoBERTa. Similar to training, the inference time of the metric model includes embedding times from MPNet which can be further reduced with smaller embedding LLMs. Overall, we can conclude that the complexity of the metric framework is significantly lower than finetuning supervised LLMs, especially in training. This advantage is also important in that it allows an effective detection system to be trained much easier in consumer hardware which means more accessibility to the general public.

4 Conclusion

The recent breakthrough in large language models, while offering tremendous potentials to society, also brought forefront the needs of cheap and effective methods to identify if contents are generated by human or artificial intelligence. With such motivation, in this paper, we focus on the task of identifying whether texts are originated by human or LLM with a light-weighted and accessible approach. For such purposes, our detection framework is trained to signify the similarities among LLM responses while boosting their dissimilarities to human-generated ones. More specifically, the framework consists of an embedding component and metric component. The embedding component is pretrained to output vector representations for raw text data. The metric neural network then further take the embedding vectors to generate their distances. Architecture-wise, we propose two metric models, one at the full-text level, and one at the sentence level. The full-text model consists of stacked feed-forward blocks, whereas the sentence model follows the transformer architecture. Both models are trained using our same-context sampling strategy designed for LLM text detection in both pairs and triplets. Experiments show that our best models obtain F1 in between $0.87 - 0.96$ in multiple settings, while offer much better time complexity than the supervised RoBERTa.

For future works, we will explore the following directions. First, we will explore more architectures for the metric components, as they are the core of this detection paradigm. Second, we will develop methods that can effectively reconstruct contexts from any texts, as this information is not always available. Besides using questions, some works have utilize a start portion of the texts themselves, which limit the use cases to longer inputs. Finally, we will adapt this work to the general cases such as modeling longer texts in the form of essays or documents, or where the generative LLMs are not known at decision-making times.

REFERENCES

Yoshua Bengio and Olivier Delalleau. Justifying and generalizing contrastive divergence. *Neural computation*, 21(6):1601–1621, 2009.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Prithiviraj Damodaran. Parrot: Paraphrase generation for nlu., 2021.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Jesus Guerrero and Izzat Alsmadi. Synthetic text detection: Systemic literature review. *arXiv preprint arXiv:2210.06336*, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Matt Gardner Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023.

OpenAI. Gpt-4, 2023. URL `https://openai.com/gpt-4`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33: 16857–16867, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. Consistency of a recurrent language model with respect to incomplete decoding. *arXiv preprint arXiv:2002.02492*, 2020.

Wikipedia, Dec 2017. URL `https://en.wikipedia.org/wiki/Category:Glossaries_of_science`.

Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.