LEVERAGING LLMS AS DOMAIN EXPERTS FOR EVALUATING VERACITY IN MACHINE GENERATED QA TEXTS

LIZ CHU, ALEX CHENG, PRATHIK GUDURI, AND KEVIN REN

Abstract. With the increase of large language models (LLMs) taking up a significant amount of space within the world of AI, there have been many questions raised about the validity of AI-generated responses. Models such as InfoGAN and GPT have made large strides in progress for generation in computer vision and natural language processing. In this project, we dive into the literature of the validity of generated responses in machine-generated text, specifically after an open-ended question prompt: given a response R from a model M based on a prompt P, how do we ensure that R is true in the actual world?

Introduction

In the past few years, there has been a significant improvement in the sphere of natural language generation (NLG) with the rise of transformers-based architectures, which has been used in several downstream tasks such as summarization, language modelling, and question answering. However, many of the metrics are based on orthographic and syntactic similarity to the gold responses rather than the content itself. Furthermore, the lack of explainability in such complex architectures makes it difficult to pinpoint features of natural language samples that point towards its truthfulness. Given the ambiguity of the task in assessing truth in machine generated text, we created a general roadmap on how we want to assess the current state of research in fact checking artificially generated responses. Specifically, we want to ask

How do we measure the truth of a machine-generated text?

Literature Review

Language Models for Open Domain Question Answering Question answering and information retrieval has been an important task in natural language processing for a significant period of time. Some earlier works focused on the use of similar n-grams, with algorithms such as tf-idf and BM-25 (Robertson and Zaragoza, 2009) being one of the first few algorithms that focused on getting the correct documents based on the number of equivalent words in the query and the document. These models would ignore semantic and syntactic features of the context, and would focus solely on exact word match similarity. Thus, the classic question answering pipeline followed the order of

(1) question processing in order to extract key features
(2) indexing to represent documents as indexed vectors
(3) passage retrieval to get relevant passages
(4) answering processing, to extract answers from passages.

This pipeline also focuses on open-domain question answering Voorhees and Tice (2000) - while many question answering models require a context to accompany the question, we are interested more in the model needing to sample a large cornucopia of documents, or have that knowledge "embedded within" the model.

Date: March 19, 2024.
CMU F23 10-701 Final Project Report

These QA models also tend to be extractive, and expect to take out exact words and phrases from the document in order to generate the response. However, another way to frame question answering is as a sequence to sequence task, where given a sequence of tokens, a model must generate a logical sequence output. This is a more abstractive approach given the generation of new sentences, while keeping the content the same. Modern encoder-decoder architectures for question answering often focus on the abstractive approach, as it is impractical to always expect an exact answer in the document text.

While there has been significant work done in past architectures, many works today focus on transformers (Vaswani et al., 2023) due to their ability to have long range dependencies that other architectures perform poorly at. For example, given the sentence

The cat hated me because I would always sit on its tail.

Architectures that suffer from the vanishing gradient issue such as the RNN or LSTM would have trouble linking "its" and "cat", but the self-attention mechanism in a transformer would facilitate this process. Thus, given the question

What hated me sitting on its tail?

The transformer would have an easier time "remembering" that it was the cat.

Thus, models such as BERT (Devlin et al., 2019), GPT (Brown et al., 2020), and T5 (Raffel et al., 2023) have achieved state of the art performance for not only trivia-based question answering tasks (who/what/where/etc.), but also more open ended questions. More in the realm of open-domain question answering specifically, there has also been work on improving step 3 of passage retrieval, with the concept of dense passage retrieval (DPR) (Karpukhin et al., 2020) leveraging the transformer approach in selecting documents using a dense embedding scheme to achieve results that surpass BM-25 and tf-idf.

Datasets for Question Answering While it is important to have models that can accurately predict the correct answer to a question, it is also of utmost importance that the training data for these models are the training data itself. The difference between vanilla question answering and open-domain question answering data is negligible, as for the latter you simply drop the context to force the model to search for the domain, so often in practice the same dataset could be used for both related applications.

One of the most influential question answering datasets is the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016), which consisted of over 100,000+ questions and answers based on a set of Wikipedia articles, with some questions in SQuAD 2.0 even being unanswerable. Other influential datasets include HotpotQA (Yang et al., 2018), which would require context from multiple documents, as well as TriviaQA (Joshi et al., 2017), which created questions before retrieving relevant documents rather than the typical approach.

Nevertheless, all of these datasets suffered from the issue of having "black and white" answers, where an answer was either clearly correct or clearly incorrect. For example, while there is ambiguity in the question "What caused my parents' divorce?", as there is no clear cut answer, no partial credit can be given for "When did WWII start?" or "What is the fastest land animal?". While the Natural Questions dataset (Kwiatkowski et al., 2019) assuages this a bit with the paragraph-long response answers, only recently has KhanQ (Gong et al., 2022) released a dataset which required answers that are more than just shallow extractive responses. Sample questions from this dataset include "How does water evaporate?" - which may incur partial credit for incomplete responses. Nevertheless, it is expensive to generate datasets with non-shallow answers, as its non-extractive nature lends it to be much harder to generate answers to deep questions.

Frameworks for Fact Checking Machine Text Some of the most common metrics that have been used to fact check machine text have been precision, recall, and F-1 score; metrics that are widely used in almost every sequence to sequence task within natural language processing, as well as other fields of science. However, these metrics aren't tailored to any features of machine text, nor natural language processing as a whole, and as such may not be the best metric to utilize. A popular metric that came out for language models was BLEU (Papineni et al., 2002), which was originally built for the task of machine translation. BLEU would find common n-gram counts between the source and predicted text for multiple n, and calculate the average weighted score, as well as a brevity penalty. While this builds upon our old metrics by adding the idea of n-grams, it also focuses on precision while discarding recall, and also does poorly with words that contain complex morphologies . Another metric that was proposed was ROUGE (Lin, 2004), which focuses eponymously on recall instead, and also follows the same n-gram approach, but just as BLEU fails when words have similar meanings or complex morphologies.

A more critical error with these metrics is that they do extremely poorly for fact checking responses that are similar in prose and word choice, but completely different in content. For example, consider the phrase

He is *not* the President of the United States.

If the correct answer was that the person in question was in fact the President of the United States, ROUGE and BLEU both would give wildly incorrect metrics, because even though there are a significant amount of common n-grams, the meaning is literally flipped.

Thus, there has been a lot of recent work in attempting to get the abstractive truth from machine generated text, rather than an extractive answer. For example, this paper, written by a group of researchers at Amazon, defines a method to evaluate whether a given generated text is true or not, called DepChecker (Estes et al., 2022). They firstly make the assumptions, for this specific evaluation setting, that there exist lots of structured data in the relevant knowledge domain, that can be used to evaluate the veracity of a given text, and also that the outputted text (to be evaluated) from the model follows some predefined structure and style as well. Due to the presumed structure of the generated text and information on hand, the researchers first tokenize the generated text, using a set of pre-defined rules to perform dependency parsing and identify the heads of entity mentions in the generated text. They then use the tokenized version of the text to identify the entirety of the entity mentioned, and connect this entity to the pool of gold labeled data to determine the veracity of the text. This work thus identifies a fast method to directly determine the veracity of machine-generated text, that was often faster than state-of-the art methods such as neural methods in such structured scenarios, with high precision on true texts and high recall on false texts. Thus, while the other papers that we include outline better metrics for evaluating veracity or methods for improving the ability of Large Language Models to output true text, this project provides a strong method for performing the actual step of evaluating veracity itself, which presumably is also more interpretable than neural methods of doing the same task due to its use of more rule-based decision making.

Furthermore, this paper from Google Brain defines a new metric to address the limitations of existing evaluation metrics like ROGUE and BLEU which primarily measure local n-gram overlap.(Goodrich et al., 2019) The authors define facts as relation tuples in the form (subject, relation, object) — for example, the statement "Barack Obama II, born August 4, 1961" can be represented as the fact tuple (Barack Obama, born on, August 4 1961). Their metric calculates the precision of factual content in generated text compared to the ground-truth text by looking at the overlapping (subject, relation) pairs and whether their (object) values match. A significant advantage of this metric, which they name "factual accuracy," is that they are able to capture fine-grained accuracy. For example, when evaluating the statements "Johnny is in NCT 127" and "Johnny is

in NCT Dream" (where the former is ground-truth and the latter is generated), factual accuracy would yield a score of zero due to the mismatch between objects, while ROUGE-1 might produce a non-zero score due to the overlap of the first four words. A challenge the authors present is that when sentences have different structures, this metric may no longer accurately represent factual equivalence (e.g., "Taeyong is the leader of NCT 127" and "NCT 127's leader is Taeyong" would be classified as two different facts, despite conveying the same information). Their proposed approaches to overcome this challenge is using either (1) Named Entity Recognition (NER) to identify and pair entities in the texts to generate their facts or a (2) transformer-based model to create an end-to-end structure that maps different sentence orderings to the same facts. This novel metric would be an interesting investigation for our paper, as it deviates from the common evaluation metrics we see with generative text. As it also addresses limitations of its proposed metric, this paper may be a good idea to investigate in tandem with other metrics that do not have the same issue (i.e., a metric for evaluating factual accuracy that is not impacted by the sentence structure of the given generated text).

Frameworks for Text Generation and Evaluation with Small LLMs In a world where computer scientists are constantly trying to add more and more parameters to LLMs in order to generate increasingly realistic-sounding text generators, this paper written by scientists at Microsoft Research focuses on generating small models - in particular, how small can an LLM, trained on simple stories at a toddler's reading level, be while still managing to produce coherent stories of their own that demonstrate strong cognitive reasoning skills and perfect grammar?

In summary, the researchers were able to find that models that were significantly simpler in terms of architecture and/or parameter counts were still able to demonstrate the ability to generate human-sounding text as well as demonstrate some ability to reason and accumulate knowledge. By using large models such as GPT-4 to evaluate the quality of the outputs of smaller models (e.g. less than 1 million parameters, only 1 attention head), all trained on the TinyStories dataset containing texts of stories directed at children, and also manually reading over arbitrary outputs of the models, the researchers were able to identify that even particularly small models (those with only 64 neurons in the hidden layer, for example) could produce relatively high-level stories.

The motivation behind investigating the strengths and weaknesses of smaller models also extend to interpretability and efficiency. In particular, the researchers argue that the fact that a model containing one attention head could still output reasonable results was promising, because in such shallow models the attention head is directly responsible for the output tokens, implying a more interpretable output function and simpler-to-analyze attention scores for individual items in the output. Furthermore, since fewer neurons exist in these models they argue that individual neurons could at times learn unique stylistic or semantic information that affected the text generation process. The fact that smaller models are by definition simpler to train than larger ones also meant that each of the tiny models took time in the order of a single day to train, a large improvement over the status quo of massive LLMs such as GPT-4.

Finally, treating model size as an ablation for larger LLMs such as GPT-4 motivated the final contribution of this paper, which was a method to evaluate the generated text of smaller LLMs using larger LLMs. Once trained on the TinyStories dataset, the smaller LLMs were then asked to generate stories of their own using a set of prompts. These stories were then passed into GPT-4, which was asked to evaluate the stories using a set of criteria such as creativity and grammar, using real numbers from 0 to 10. This paper thus outlined a framework for generating robust metrics to evaluate LLM output using larger models as domain knowledge experts. We hope to draw upon this segment of the contributions in order to take advantage of the convenience and flexibility, in

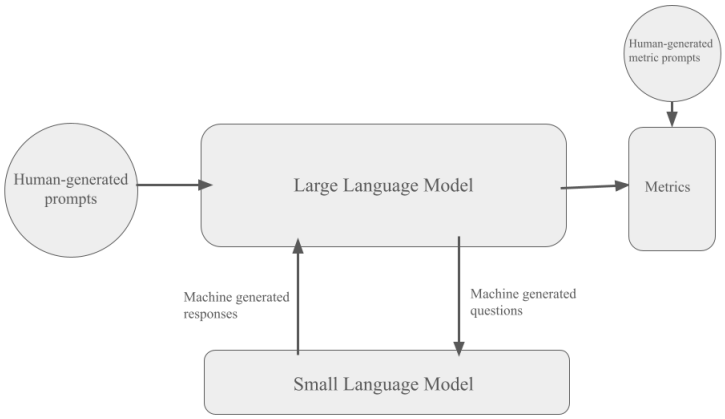

FIGURE 1. Testing Pipeline

Model Name	Number of Parameters	Ref
GPT2-medium	380M	link
GPT2-xl	1.61B	link
EleutherAI/GPT-neo	1.37B	link

FIGURE 2. Small Model Parameter Numbers and Links

terms of deriving novel metrics, that large model evaluations have in the context of assessing the factuality of machine-generated text in smaller models. This development will offer an answer to our initial question of what defines truth verification with LLMs.

However, we must be cautious with our use of LLMS as a form of a paradoxical distant ground truth, as they may be "narcissistic" (Liu et al., 2023). More specifically, many prominent LLM-based scores, such as BARTScore, T5Score, and GPTScore have shown to have a latent bias towards their own respective architectures when assessing the validity of machine generated text. Thus, without gold reference summaries, we must tread carefully when using these metrics and perhaps provide consistency in model choice or control for it when conducting experiments.

Experimental Results

Methods The goal of our paper is to show the power of using large language models to evaluate the accuracy and truthfulness of a smaller language model. To do this we will perform a demonstration of this technique on specific models and a dataset. We will include two models, one smaller language model, and one larger model. We will give the smaller language model inputs from the chosen dataset and then provide the large language model with the input and also the output of the smaller language model. We will also request certain metric values from the larger language model that measure how good of an answer the smaller language model provided.

The ability to use a larger model to accurately evaluate a smaller model would be very useful for training the smaller model at rapid speeds by leveraging the knowledge of the larger model. This can help train smaller models that have more specific purposes than larger more general models (e.g. ChatGPT). Further, these methods will just become more and more applicable as these large languages rapidly become more and more powerful as they have been in recent years.

We chose three "smaller" models to test, with their names and number of parameters highlighted in Figure 2. We use GPT 3.5 as our domain expert for checking the outputs of our smaller models, as well as our open-ended question generator. With GPT 3.5, we created 150 open-ended questions that had topics ranging from scientific discoveries to historical events. An important note of feeding these questions in the smaller model is that as the transformers pipeline requires context for QA tasks, we framed each question as a text-generation task with the prompt "Question: [question to answer] Answer: " so that each model will fill out the response. We limit each small model's response to between 50 and 100 tokens, which upon manually observing results, we found that most responses had sentences cut off by this 100 token limit.

After generating 150 responses for each of our small models for a total of 450 responses, we use the ChatGPT-3.5 API to ask our domain expert to rate each of the responses on an integer scale of 1 to 10 in each of the four categories: Grammar, Accuracy, Relevance, and Completeness. Below is the prompt that we prepend to each response.

> I have a pair of question and answer and I would like you to rate the answers. The answer provided may be nonsensical and even contain characters that just do not make sense. If this is the case, then rate the response to the best of your ability. Do not come up with hypothetical answers if the question is unanswered. I want you to rank the responses on an integer scale of 1-10 in the categories of grammar, accuracy, relevance, and completeness and give a one sentence explanation for each. Your answer should specifically be formatted as "Grammar score: explanation Accuracy score: explanation Relevance score: explanation Completeness score: explanation" with a new line between each of the categories in the order of grammar, accuracy, and completeness. Please do not repeat the input given to you in your answer and make sure all scores are created independently of each other.
>
> The input format bellow will be "Question: [prompt for the answer] Answer: [response to grade]". There may exist the word "Question" and "Answer" in the response for you to grade, but consider those as part of the response. The entire response you should grade starts after the first instance of "Answer:". Below is the input:

Discussion and Analysis

Analysis of Results Overall, it seemed that the larger XL version of GPT-2 dominated the smaller medium iteration: for the metrics that we asked GPT-3.5 to rank the smaller model output using, GPT-2 XL had consistently more frequent occurrences of ratings 7 through 10, and lower frequencies of ratings 1 through 6.

One significant thing we notice through the ratings is that the models that do better than the others in one category generally also seem to do better in the other categories. This would make sense as often the better and bigger models just do better in all of these categories.

A significant exception to the above pattern is the grammar performance between the medium and XL models. The grammar ratings between medium and XL follow pretty similar curves on the histogram with averages of 4.33 and 4.49 respectively. However, when it comes to the other three categories, we observe that the XL model does significantly better than the medium model. We believe that this could be due to certain confounding factors that appeared because of issues we encountered. One of these issues was the models outputting random nonsensical characters that ChatGPT thought lowly of for Grammar score, and thus potentially brought down the score of both models to similar values. The other was the fact that we cut the answer of the models off at 100

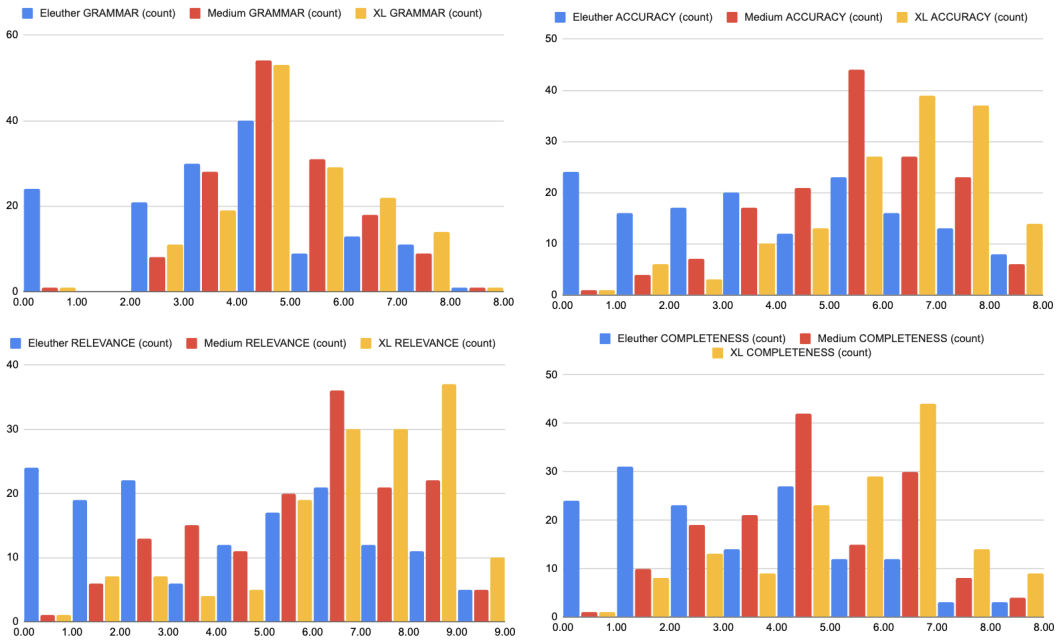

FIGURE 3. Distribution of Accuracy, Completeness, Grammar, and Relevance. Note that zeros represent nonsensical responses, or ones where GPT 3.5 was unable to provide a rating.

Model Name	Grammar	Accuracy	Relevance	Completeness
GPT2-medium	4.33	4.96	5.37	4.15
GPT2-xl	4.93	5.56	6.16	4.88
EleutherAI/GPT-neo	3.36	3.52	3.74	2.71

FIGURE 4. Average ratings for each model

words due to how the models often just kept generating, thus this limits the ability of the "better" model to perform better in grammar.

Another thing that we found very interesting is that the Eleuther model did significantly worse than the other models even though it had more parameters then the medium GPT-2 model. Consider the following Eleuther question-answer pair, along with the feedback from GPT-3.5:

"Question: Can you explore the challenges and opportunities in balancing economic development with environmental conservation? Answer: _____

To continue the conversation, please email your feedback, questions and thoughts in a comment below.

The original email to: _____
The original email to: _____
The original message was published on: _____
The original message was published on: _____
The original message was published on: _____"

As we can see this answer does not make any sense which is reflected in GPT-3.5's ratings of

2, 1, 1, 1 for grammar, accuracy, relevancy, and completeness respectively. This is a common occurrence in the answers produced by the Eleuther model. This could have many reasons, one potential one is the difference in the training corpus between GPT-2 and Eleuther.

Shortcomings and Future Work Several limitations or constraints potentially apply to our conducted experiment. For instance, one characteristic of LLMs such as GPT-3.5 is their stochastic behavior given identical prompts: provided the same input texts, successive calls to the LLM may result in different or sometimes completely different outputted texts. Future work or replications of this work could take this fact into account by rerunning GPT-3.5 (or whatever largest LLM is being used as the domain expert) multiple times on each question-answer pair returned by the smaller LLMs, and taking an average over these individual ratings to get the average rating per question-answer pair. This could potentially account for some of the variability of LLM responses from the grader perspective.

Secondly, the confounding factor of non-UTF-8 characters appearing in our smaller LLM outputs along with cut-off sentences represents unexpected behavior that may have influenced our across-the-board lower grammar scores. These lower grammar scores may also have influenced lower ratings across the board for other categories as well. This clearly was a major problem for the Eleuther model, where oftentimes the model had nonsensical outputs in response to unrelated questions (e.g responding to an email when the question was about the relationship between economics and the environment). The challenge of applying our evaluation framework to small LLMs that still are able to consistently produce coherent responses is a step which future projects must understake.

This work finally enables the use of LLM-as-domain-expert feedback ratings to be used as true labels of variables such as veracity, grammar, and accuracy, which could ultimately be used to fine-tune or improve existing text generation or question-answering models. Our framework represents a flexible method for using domain experts such as LLMs to evaluate the work of smaller models using any arbitrary number of user-provided metrics as feedback - using these question-answer pairs along with the domain expert feedback as a label could be a fascinating way to fine-tune models in scenarios where evaluating true values of the metric to develop a supervised learning task could be a painstaking task involving human annotation (or altogether infeasible).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alex Estes, Nikhita Vedula, Marcus Collins, Matthew Cecil, and Oleg Rokhlenko. 2022. Fact checking machine generated text with dependency trees. In *EMNLP 2022*.

Huanli Gong, Liangming Pan, and Hengchang Hu. 2022. KHANQ: A dataset for generating deep questions in education. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5925–5938, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023. Llms as narcissistic evaluators: When ego inflates evaluation scores.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

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

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

5000 Forbes Ave. Pittsburgh, PA 15213
Email address: echu2@andrew.cmu.edu

5000 Forbes Ave. Pittsburgh, PA 15213
Email address: abcheng@andrew.cmu.edu

5000 Forbes Ave. Pittsburgh, PA 15213
Email address: pguduri@andrew.cmu.edu

5000 Forbes Ave. Pittsburgh, PA 15213
Email address: kevinren@andrew.cmu.edu