First Progress Report

Yichen Huang(yh3550)
Taichen Zhou(tz2555)
Cong Chen(cc4887)
Ruolan Lin(rl3312)
Longxiang Zhang(lz2869)

October 20, 2023

Contents

1 Introduction

The evolution of Large Language Models (LLMs) represents a monumental shift in the landscape of artificial intelligence. These models, equipped with billions of parameters, have transformed from mere tools of text prediction into sophisticated engines capable of understanding, generating, and even reasoning with human-like prowess. The applications of LLMs have permeated numerous sectors, but their potential in the financial industry stands out as particularly promising—and challenging.

Financial data, by its nature, is intricate, nuanced, and laden with domain-specific jargon. The implications of decisions made based on this data are profound, affecting everything from individual investments to global economic trends. As LLMs are increasingly tasked with roles such as document summarization, risk assessment, and regulatory compliance assistance, the margin for error narrows. It is not enough for an LLM to produce coherent content; it must also be factually accurate, contextually relevant, and devoid of misconceptions.

Recognizing this challenge, Fidelity Investments has embarked on an ambitious initiative. The goal is clear yet complex: to craft a evaluation framework for LLMs tailored to the intricacies of the financial sector. Such a framework seeks to move beyond traditional metrics of text coherence or fluency. It delves deeper, probing for correctness in the face of financial terminologies, sensitivity to nuanced prompts, and the LLM's ability to reason without falling into the trap of 'hallucination'—generating content that sounds plausible but is factually incorrect or baseless.

Our collaboration with Fidelity Investments, facilitated by the Capstone Project initiative, positions us at the forefront of this endeavor. We are not just engineers or data scientists in this journey; we are pioneers, navigating the confluence of finance and AI. Our mandate is to ensure that as financial institutions increasingly rely on LLMs, they do so with a clear benchmark of reliability and accuracy.

This report chronicles our ongoing efforts, offering insights into our methodologies, the rationale behind our choices, and the iterative process of refining our evaluation framework. As we delve deeper into the project, we remain cognizant of its broader implications: establishing a gold standard for LLM evaluations that could guide industries beyond finance, ensuring that as the world leans more into AI-driven solutions, it does so with informed confidence and rigor.

2 Phase I: Initial Setup and Problem Definition

The first step was decisive when we embarked on this ambitious project to design an assessment framework for lifelong learning programs in the financial sector. This phase laid the foundation for the entire project. It was characterized by intense consultations, brainstorming sessions, and careful planning. It was imperative to align our vision, understand the nuances of the challenge, and establish a solid logistical framework before delving into technical details and specific methodologies. Initial consultations were at the heart of this preparatory phase, during which the contours of the project began to take shape.

2.1 Logistical Setup

2.1.1 Tool Selection

In today's dynamic data science environment, the strategic choice of tools can significantly influence the trajectory of a project. Recognizing this, our team made judicious selections that harmoniously combined programming languages, containerization platforms, and cloud services. A cornerstone of our toolkit is **Python**, which stands out as a favorite among data scientists and machine learning enthusiasts. Its rich ecosystem of libraries, tailored to our project's requirements, provides both versatility and power. To ensure consistency across our team's diverse development environments, we embraced **Docker**. Its containerization capabilities allow the creation, deployment, and execution of applications in uniform containers, mitigating the challenges posed by disparate local setups. Lastly, for projects demanding considerable computational resources, particularly those involving Large Language Models, cloud platforms are indispensable. We opted for **GCP (Google Cloud Platform)**, drawn by its renowned reliability, scalability, and an array of tools optimized for machine learning and data processing tasks.

2.1.2 Git and Version Control

To ensure seamless collaboration and maintain a traceable workflow in our data-driven project, we adopted Git, the premier distributed version control system, complemented by GitHub's platform capabilities. Our dedicated GitHub repository centralized all code and documentation. A structured branching strategy was employed, with feature branches for specific tasks, which, post rigorous review via pull requests, were merged into the main branch, ensuring the codebase remained organized and up-to-date. This approach, fortified by GitHub's tools for communication, planning, and documentation, established a robust, transparent, and efficient framework for our project's logistics.

2.2 Problem Definition and Understanding

2.2.1 Project description

Fidelity Investments proposes a project to develop an evaluation framework for LLMs in the context of the financial industry. The aim is to quantify key facets like correct-

ness, sensitivity, and reasoning in LLM-generated content, focusing on financial and regulatory jargon.

2.2.2 Our Understanding

As we delved into the intricacies of this project, our primary objective crystallized: to architect a robust methodology that evaluates content generated by LLMs. This methodology wouldn't merely assess the content's coherence but would also rigorously gauge its accuracy and contextual relevance. To complement this, we envisioned an automated system, finely tuned to measure and score LLM outputs using a predefined set of metrics.

Venturing further into our aspirational goals, we aimed to develop a network graph that intricately maps the relationships between various entities. Such a visual aid would illuminate the intricate interplay between these entities, offering deeper insights into their interactions.

Our system's design was straightforward in terms of input and output. It would ingest content produced by an LLM and, in return, deliver a detailed evaluation of the LLM's response. This evaluation would encompass scores across pivotal dimensions, such as the content's correctness, its reasoning prowess, and the variation evident in the generated narratives.

2.3 Literature Review and Preliminary Research

2.3.1 Academic Paper

A comprehensive understanding of the current state of research on the evaluation of language models, especially with respect to their factual consistency, was essential for our project. We delved into several academic papers to gain insights into prevailing methodologies, challenges, and innovations. Three key papers stood out, providing foundational knowledge that informed our approach:

Factuality Enhanced Language Models for Open-Ended Text Generation by Nayeon Lee *et al.* from the Hong Kong University of Science and Technology and NVIDIA [1]. This paper highlighted the vulnerability of pretrained LMs in generating non-factual information. It introduced the *FACTUALITY PROMPTS* test set and metrics for assessing the factuality of LM outputs, noting the increased factual accuracy in larger models.

Evaluating Factuality in Generation with Dependency-level Entailment by Tanya Goyal and Greg Durrett from The University of Texas at Austin [2]. This research proposed a novel method for evaluating factuality at the level of dependency arcs. By focusing on the semantic relationships in generated content and their alignment with source data, it provided a granular approach to fact-checking.

Evaluating the Factual Consistency of Abstractive Text Summarization by Wojciech Kryściński *et al.* from Salesforce Research [3]. This paper underscored the limitations of current summarization assessment metrics, especially in terms of factual consistency. A weakly-supervised, model-based approach was proposed to verify the factual consistency of summaries and identify discrepancies with source documents.

2.3.2 Industry Examples

The world of academic research is a treasure trove of theoretical foundations and cutting-edge methodologies. Yet, the real test of these concepts lies in their application within the industry, where unique challenges and practical constraints come into play. In the financial realm, this interplay between theory and practice is vividly exemplified by two industry giants: Bloomberg and Deutsche Bank. Both have ventured into the domain of LLMs, and their journeys offer invaluable insights.

Bloomberg, a name synonymous with financial data and analytics, has always been at the forefront of innovation [4]. Their exploration into LLMs was no exception. They embarked on a rigorous evaluation journey, scrutinizing LLM outputs across a gamut of tasks, both financial and general. Specific metrics, such as FPB, FiQA SA, Headline, and NER, became their compass, guiding them towards precise evaluations manifested in F1 scores and exact match accuracies. Notably, their internal Sentiment Analysis system became a beacon of their commitment to nuanced evaluations. This exhaustive approach culminated in the birth of *BloombergGPT*, a testament to their dedication to refining LLMs for financial expertise.

On the other side of the spectrum, Deutsche Bank, a stalwart in global banking, charted its unique path in the LLM landscape [5]. Their approach was holistic, viewing LLMs not just as computational entities but as tools with immense potential for the financial industry. Validation accuracy, real-world applicability, and tasks like document summarization became their milestones. Furthermore, they championed the cause of model interpretability and explainability, ensuring that their LLMs were not just black boxes but comprehensible entities. Their rigorous stance against adversarial inputs and their adaptability to dynamic financial scenarios underscored their vision. The outcome? A dedicated LLM named *Large Language Models in Finance*, embodying Deutsche Bank's ethos and commitment to excellence.

In synthesizing these industry narratives, one thing becomes clear: while academic research lays the groundwork, it's the industry's practical implementations that bring these concepts to life. The endeavors of Bloomberg and Deutsche Bank validate the significance of LLM evaluations, offering a lens into the challenges, triumphs, and nuances that academic corridors might sometimes overlook.

3 Phase II: Data Wrangling and EDA

The primary objective of our project involves working with raw data extracted from final litigation files, which are primarily in PDF format. Our initial step in this endeavor was to download these PDF files and attempt to extract the text data contained within them. However, we encountered challenges when using the PyPDF2 package for text extraction, as the results were less than satisfactory. We observed numerous word mistakes and line mismatches in the extracted text, which posed a significant obstacle to our goal of creating robust evaluation metrics.

As a result, we turned our attention to Optical Character Recognition (OCR) technology. OCR, which stands for Optical Character Recognition, is a technology that facilitates the electronic or mechanical conversion of images containing typed, handwritten, or printed text into machine-encoded text. This process can be applied to scanned documents, photographs of documents, or even text embedded within images, such as signs and billboards in landscape photos or subtitles from television broadcasts. In our quest to enhance the quality of text extraction from PDF files, we adopted the Pytesseract library, which is a Python wrapper for Google's Tesseract-OCR engine. By applying Pytesseract to the PDF files, we were able to extract the raw text data more effectively than when using PyPDF2. The results obtained through this OCR approach exhibited improved accuracy and fidelity, addressing the word mistakes and line mismatches that had previously hindered our progress. This advancement marked a significant milestone in our project, as it enabled us to work with cleaner and more reliable text data.

After applying OCR to extract text from our documents, one of the most common issues we encountered was the presence of unnecessary line break characters, often represented as '\n'. These line breaks, while seemingly innocuous, could potentially impact the performance of our Large Language Model in the subsequent steps, particularly when generating summaries. To address this concern, we implemented a solution to manage and eliminate these extra line breaks, thus enhancing the overall quality of our text data for the subsequent summarization process.

Additionally, we recognized that the identifications appearing on every page, such as the example you provided, "\nCase 1:10-cv-11665 Document1 Filed 09/29/10 Page 2 of 18\n," could also adversely affect the quality of the generated summaries. These identifiers are often extraneous and carry no informational value for our summarization task. To remedy this issue, we leveraged regular expressions to automatically detect and remove these page identifiers whenever they appeared within the text files.

4 Phase III: Initial Modeling and Framework Design

In the process of selecting large language models for the purpose of generating summaries, our study involved the evaluation of three prominent models: 'facebook/bart-large-cnn' sourced from Hugging Face, ChatGPT 4, a well-established model, and Claude which is called a potential "ChatGPT killer." All of these models were employed to produce summaries for the evaluation phase of our research. During this phase, our dataset consisted of a limited set of 31 reference summaries, each matched with its corresponding source text. Consequently, the summaries generated were based exclusively on this set of 31 source texts.

Prior to embarking on an exploration of the framework employing dependency arcs for the assessment of the factual accuracy of large language models, we present a conventional approach for benchmarking purposes. In this context, we have opted for established metrics, including Recall-Oriented Understudy for Gisting Evaluation (ROUGE), cosine similarity, Metric for Evaluation of Translation with Explicit ORdering (METEOR), and Bilingual Evaluation Understudy (BLEU).

4.1 Metrics Introduction

4.1.1 ROUGE

ROUGE is a metric that evaluates text summarization by comparing overlapping n-grams between generated and reference text. There are many types of ROUGE scores:

ROUGE-N: measures the number of matching n-grams between the generated text and the reference text.

ROUGE-L: evaluates the longest common subsequence between the generated text and the reference text.

We choose ROUGE 2 F1 score as a metric in our baseline.

4.1.2 Cosine Similarity

Cosine similarity is a metric for measuring the similarity between two vectors, often used to assess the similarity between texts in a multi-dimensional space. It is very important to choose the embedding for Cosine Similarity since we need to embed each summary into vector first then calculate the cosine similarity value based on the vectors. Here we choose the classical TF-IDF embedding.

4.1.3 METEOR

METEOR is a machine translation evaluation metric that assesses the quality of machine-generated translations by measuring the similarity and fluency of the generated text compared to the reference text. It takes into account several factors, including unigram precision, recall, stemming, synonymy, and word order, to provide a more comprehensive assessment of translation quality.

4.1.4 BLEU

BLEU score is usually used alongside metrics like ROUGE and METEOR to assess the text generation. It emphasizes precision in n-grams between the generated text and the reference text.

4.2 Result Analysis

We employ conventional evaluation methods to assess the three aforementioned large language models. In the case of each model, we initiate the process by inputting source documents into the model to generate summaries. Subsequently, we conduct a comparative analysis between the generated summaries and manually crafted reference summaries by professionals. The evaluation scores are then calculated using established approaches. The results table is presented below.

Model	ROUGE2 F1 Score	Cosine Similarity	METEOR	BLEU
Facebook Bart	**0.294**	0.527	0.298	**0.162**
ChatGPT 4	0.235	**0.541**	**0.353**	0.161
Claude	0.213	0.499	0.291	0.113

In the presented table, we have exclusively considered the average scores obtained from the evaluation of all 31 generated summaries. An examination of these results reveals variations in the performance of different models across various evaluation methods. According to the ROUGE and BLEU metrics, our analysis consistently identifies 'Facebook Bart' as the model achieving the highest average score. This implies that 'Facebook Bart' exhibits superior performance in terms of summary quality according to these particular metrics. However, the landscape changes when employing the 'Cosine Similarity' and 'METEOR' methods. Under these metrics, 'ChatGPT 4' emerges as the prevailing model, yielding significantly higher METEOR scores in comparison to the other two models.

In addition, we have included box plots illustrating the performance of the three large language models for each score. These visual representations offer a more comprehensive perspective of the data. Upon examining the plots, we observe that the conclusions align with our previous findings. Specifically, the ROUGE graph underscores the superior performance of the 'Facebook Bart' model, while the METEOR graph highlights the significantly higher scores achieved by the 'ChatGPT 4' model in comparison to the other two. Furthermore, the Cosine Similarity and BLEU graphs indicate a relatively minimal distinction between the 'Facebook Bart' and 'ChatGPT 4' models.

4.3 Limitations

Upon a comprehensive analysis of the implementation process and the examination of the results, it becomes evident that the baseline traditional approaches exhibit numerous limitations.

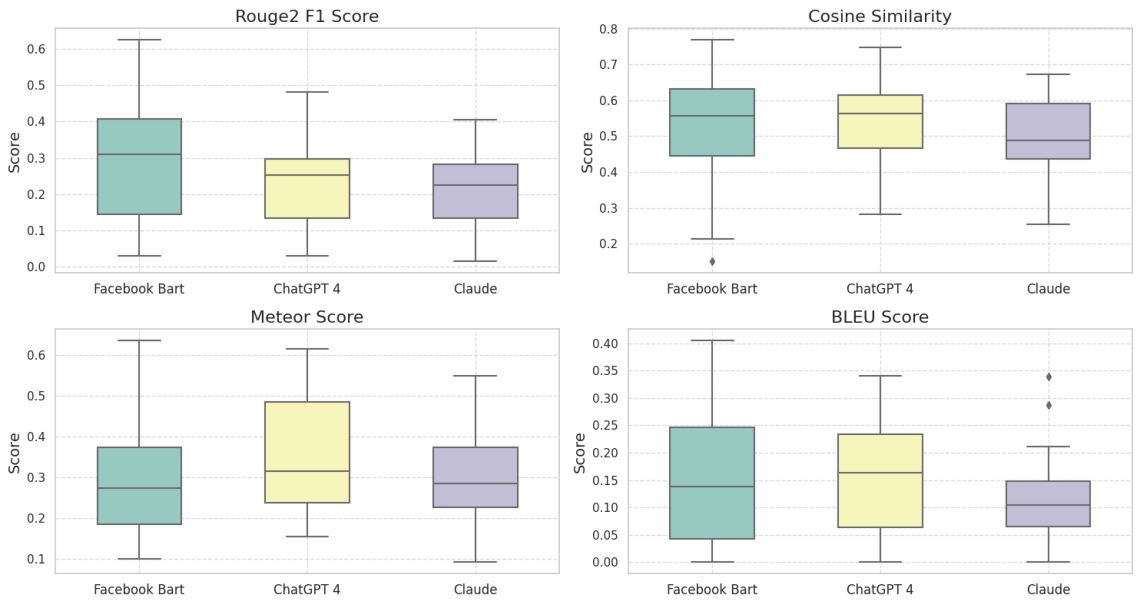

Figure 1: Box-plot for Scores

4.3.1 Data Gaps and Omissions

While a summary may receive a high score according to traditional approaches, it may still not qualify as a comprehensive summary due to the omission of vital information.

Reference Summary	Generate Summary	Score
FINRA found the firm failed to reasonably supervise for potentially manipulative trading from 2013 through 2019 and SageTrader also failed to establish and implement AML policies and procedures reasonably expected to detect and cause the reporting of suspicious activity and violated FINRA Rules 3310(a) and 2010. As a result, SageTrader consented to a censure and a $100,000 fine to FINRA. Additionally, SageTrader also had to pay a total fine of $775,000 which was paid to FINRA and eight exchanges, and the firm was also required to review and revise its supervisory system for detecting potentially manipulative trading by the firm's customers.	SageTrader has been a FINRA member since May 2006. FINRA found that the firm failed to reasonably supervise for potentially manipulative trading from 2013 through 2019. The AWC imposed a censure and a fine of $775,000 (paid to FINRA and eight exchanges) The firm considered two of the four customers to be high AML risk. The firm's market participant identifiers (MPIDs) were used to route orders to anAlternative trading system and to exchanges. SageTrader failed to establish and implement AML policies and procedures reasonably expected to detect and cause the reporting of suspicious activity.	0.769

For instance, consider the following example: We employed the Cosine Similarity method to assess a summary generated by the 'Facebook Bart' model. Despite achieving the highest Cosine Similarity score among all 93 generated summaries, this particular summary fails to encompass crucial details, such as FINRA Rules 3310(a) and the $100,000 fine. These details hold significant importance within the context of the source document, particularly within the financial domain.

4.3.2 Threshold Determination Dilemmas

In the context of each traditional approach, the selection of an appropriate threshold is imperative to assess the quality of a summary. A threshold serves as a specific criterion that determines whether a summary is deemed satisfactory or not. To illustrate, in the

case of the ROUGE2 score, designating a threshold value of 0.7 allows us to classify summaries as either "good" if their score exceeds 0.7 or "bad" if it falls below this threshold.

Nevertheless, the task of threshold selection presents considerable challenges. On one hand, despite the scores falling within the [0,1] range, their distributions exhibit significant disparities. For instance, the ROUGE2 score's highest 5% value threshold is 0.481, while the Cosine Similarity score's equivalent threshold is 0.721. Consequently, a simplistic approach of selecting a high threshold like 0.9 is infeasible due to the divergence in score distributions.

On the other hand, determining an appropriate threshold based on the distribution of scores for each traditional approach necessitates gathering scores from multiple models, which may introduce bias and one-sidedness. Additionally, reference summaries frequently approach the maximum score of 1 in most conventional methods. Therefore, selecting a threshold based on score distribution, even if relatively low, implies significant dissimilarity between the considered "good" summaries and the reference itself.

4.3.3 The Necessity of Reference Summaries

In all conventional methodologies, the inclusion of a reference is indispensable as it enables the comparison of generated summaries against the reference to determine their quality. Omitting a reference renders these methods unable to calculate a score for evaluation. However, in real-world scenarios, these models are deployed to aid in the evaluation of AI-generated summaries. Consequently, if we supply a human-authored reference, the role of the model becomes redundant, as the reference alone suffices to assess the quality of the summaries.

5 Proposed Evaluation Framework (Refinement based on Phase 3)

The shortcoming of traditional metrics evaluating the quality of summaries according to the original text is that they are usually based on the high dimensional representation of both summaries and source text so that even though they can calculate the similarity of summaries and original text, they cannot explain the calculation result in detail and in an easy understanding way. More specifically, they lack the ability to point out the exact point where a summary does not work well while evaluating it.

In light of the consideration mentioned above, Goyal et al.[2] proposed a new kind of framework to evaluate the quality of the candidate summaries in a more trackable and understandable way. In detail, a good summary should include as many of the relationships between some entities in the original document as possible and avoid including the relationships not appearing in the source text. That is to say, to some extent, the ratio of useful dependencies of the entities in the candidate summary is a good indicator of its quality. If most of the relationships of entities in a summary do appear in the source document, the summary should be regarded as high-quality, and vice versa.

5.1 Framework Structure

Based on the above discussion, we break down the framework into three parts. The first part is dependencies extraction, the second one is dependencies classification, and the last one is calculating the score by aggregation.

5.1.1 Dependencies Extraction

In the first stage, our framework applies stanza, which is an open-source Python NLP library from Stanford University, to extract dependencies in the candidate summaries. Meanwhile, the original document and the summaries are concatenated together and then inputted to BERT to get a set of word-level representations of the text. Each representation vector contains not only the information of the corresponding word but also the context information of it. Then we can use the representation set to gain the representations of the extracted dependencies. Those representations will be the input data for the classifier of our framework in the second stage.

5.1.2 Dependencies Classification

Secondly, after extracting the dependencies in the candidate summary and vectorizing them, the representations of those dependencies will be fed into an MLP classifier which is used to predict whether or not the input dependency belongs to the original text. Since the representation of the dependencies has information on the consisting entities, the dependency type, and the context information, the information in the input data is enough to train an MLP to handle the classification task with an acceptable performance.

5.1.3 Calculate the Score by Aggregation

Once the classifier classifies all dependencies in the candidate summary, we can calculate the ratio of the useful dependencies, which are classified as $f(d) = 1$, meaning that the classifier thinks the dependencies appear in the original document, to represent the quality of the candidate summary. The formulation of the metric is below:

$$s = \frac{\sum_{d \in d(h)} f(d)}{|d(h)|}, \quad f(d) \in \{0, 1\} \tag{1}$$

where $d(h)$ is the set of dependencies in the candidate summary h. The maximum of s is 1, meaning that all dependencies in the candidate summary can be found in the original text; the minimum of s is 0, meaning that all dependencies in the candidate summary are unrelated to the source document.

5.2 Data Preparation and Augmentation

Our main job for this framework is to train an MLP that can determine whether or not a dependency belongs to the source document, given the dependency representation with the context. Therefore, we need to prepare a dataset with the dependency representations as the inputs and whether or not such dependencies are in their corresponding original text as the labels.

5.2.1 Data Preparation

To get the embedding of dependencies in the candidate summary, we do the following steps: **Firstly**, extract dependencies from the reference summary (we regard the reference summary as the perfect summary). **Secondly**, get the word-level embedding for the reference summary and original text by BERT. **Thirdly**, concatenate the embeddings of both the tail and the head of dependencies and the embedding of dependency types as the representations of dependencies. **Finally**, Label those dependencies as positive since they are from the reference summary.

There are two problems for the data preparation process, one is that we only get the positive samples for training; the other is that even for positive samples, we do not have enough reference summary for obtaining them. Thus, we have to apply some tricks to augment our data.

5.2.2 Data Augmnetation

We augment our data from two perspectives. The first one is to increase the number of positive samples; the second one is to generate negative samples. As for the first part, we proposed two strategies:

Re-translation: Kryscinski et al.'s work[3] shows re-translation is a feasible way to generate new text with the same meaning as the original one. Therefore, we plan to use some state-of-the-art translation models to translate our reference summary into a mainstream language and then translate them back. After re-translation, some structures and words of the sentences in the reference will be slightly changed. We regard dependencies in the re-translated summary as positive samples.

Re-write by LLM: Another feasible method is to apply powerful LLMs such as GPT to rewrite the reference summary with the well-designed prompt. We regard the re-written summary as the reference summary so that the dependencies in it should also be the positive samples.

As for the second part, we employ LLMs to generate a set of summaries and order them based on the beam score. A lower beam score implies a poorer quality of the summary. We mark those summaries as $h_1, h_2, \ldots, h_n$ as the quality of them decreases. Then we can extract dependencies from those generated summaries. For those dependencies appearing in $h_i, i > 1$ but not in the reference summary h^*, we label them as negative. For the remaining dependencies, we leave them unlabeled and do not use them during training. The whole labeling process can be shown as the following formulation:

$$y_i = \begin{cases} 1 & \text{if } d_i \in d(x) \cup d\left(h^*\right) \\ \text{not labeled} & \text{if } d_i \in d\left(h_1\right) \backslash d(x) \cup d\left(h^*\right) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where x is the original document and h^* is the reference summary.

6 Challenges

A significant obstacle in our project revolves around the scarcity of human-generated summaries. To mitigate this challenge, we are implementing various data augmentation techniques to enhance the quality and quantity of available summaries. However, our primary training objective centers on the extraction of dependency arcs. To gauge the effectiveness of our augmentation methods, we are closely monitoring the similarity between the dependency arcs derived from human-generated summaries and those from samples generated by a Language Model (LLM).

If the similarity between the dependency arcs obtained from human-generated summaries and the LLM-generated samples is too high, it suggests that the augmentation methods may not yield the desired results. This would raise concerns about the validity of our data augmentation strategies and prompt us to revisit and refine our approach to ensure that the LLM-generated samples diverge adequately from the human-generated summaries.

7 Conclusion and Next Steps

In the coming weeks, we will delve into the training and tuning of our evaluation model, which involves the following steps. First, based on the provided documents, we will utilize LLMs to generate a set of summaries with different beam scores. In the meantime, We will enhance the provided reference summary by employing the re-translation and re-written methods. Second, with the aid of Stanza, we'll extract dependency relationships from those texts, and leverage BERT to obtain representations of these dependencies. Third, we will train an MLP to determine whether a given dependency appears in the original related document based on its representation. Ultimately, we will use the trained model to assess the quality of summaries produced by LLMs and determine the model's effectiveness.

References

[1] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34586–34599. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/df438caa36714f69277daa92d608dd63-Paper-Conference.pdf.

[2] Tanya Goyal and Greg Durrett. Evaluating factuality in generation with dependency-level entailment. *arXiv preprint arXiv:2010.05478*, 2020.

[3] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online, November 2020. Association for Computational Linguistics. URL: https://aclanthology.org/2020.emnlp-main.750, doi:10.18653/v1/2020.emnlp-main.750.

[4] Bloomberg. Bloomberg's llm evaluation and implementation, 2023. URL: https://www.arxiv-vanity.com/papers/2303.17564/#S5.

[5] Deutsche Bank. Deutsche bank's exploration into large language models, 2023. URL: https://www.nvidia.com/en-us/on-demand/session/gtcspring23-s51160/.