Controlled Entity-Centric Summarization Large Language Model

Alex Lichtenberg*, Lei Hamilton†
*DAF-MIT AI Accelerator, †MIT Lincoln Laboratory
*Cambridge, MA, †Lexington, MA

***Abstract*—Summarizing long text or corpora of texts is a critical use-case for Natural Language Processing (NLP) technology. It allows readers to parse key information from more source documentation than they otherwise could have read in a given amount of time. In contrast to generic methods, a summary that can be constrained to specific content of interest to a reader extends the usefulness of summary methods to scenarios where the entire source document is not relevant. This can be referred to as a controlled summary. This paper proposes multiple methods to use Large Language Models (LLMs) for producing summaries focused on specific entities found in a text. Our best-performing method, which we term Controlled Entity-centric Summarization LLM (CESL), uses an instruct-tuned LLM (GPT-4 Turbo) and outperforms previous state-of-the-art approaches without additional fine-tuning. We include many additional experiments in this paper to act as an applied survey for various prompting and generation strategies in the task of entity-centric summarization. We propose additional metrics for abstractive summarization performance beyond commonly used summarization metrics such as ROUGE or BERTscore and demonstrate a framework for how they can be used to identify problematic results responsibly and proactively, allowing a human in the loop to focus on review of high-payoff results. We publicly release an improved version of the entSUM benchmark dataset. We also extend the findings of previous work regarding limitations of prompting LLMs to show that given certain prompts, some LLMs will default to answers found in their parametric memory even if explicitly instructed to rely on retrieved context.[1]**

***Index Terms*—Natural language processing, controlled summarization, large language model, named entity recognition, retrieval augmented generation, metrics, prompts**

I. Introduction

SUMMARIZING long text or corpora of texts is a critical use-case for Natural Language Processing (NLP) technology. Effective summaries allow readers to parse key information from more documents than they otherwise could have processed. Methods for summarizing text can broadly be grouped into two categories: extractive and abstractive summarization [1]. Extractive summarization involves selecting the most salient text from the original document to produce a representative summary, but may not combine the selected sentences in a way that is coherent. Abstractive summarization creates a probabilistic generation of a summary that recreates the semantic meaning of the original text while maintaining coherence, but may hallucinate or use synonyms in places where exact phrasing matters, which can be problematic in certain use cases. An ideal summary tool would be able to combine elements of both methods.

Existing summarization methods have focused primarily on entire documents [1], but there are use cases that require summaries focused on specific entities or topics contained in a document. We refer to the creation of a summary that focuses only on the content relevant to a single entity of a user's choice as an entity-specific summary.

In this paper we explore multiple methods leveraging Large Language Models (LLMs) to perform entity-specific summarization, acting as an applied survey of prompting and generation strategies on the task. Our highest performing method, which we term Controlled Entity-centric Summarization LLM (CESL), outperforms previous state-of-the-art results using an instruct-tuned LLM without requiring additional fine-tuning. We also propose additional ways to measure the quality of generated summaries and provide a framework to help enable a responsible deployment of LLM-driven summary models. Lastly, we publicly release an improved version of the EntSUM dataset to enable future work.[2]

II. Background and Related Works

Summarization techniques have benefited greatly from recent advances in NLP such as the transformer [2] and the BERT architectures [3], opening new opportunities for research into sub-fields like controlled abstractive summarization, which allows readers to tune parameters like reading level and length [4]. Our work extends this idea and focuses on generating a summary specific to the context relevant to a given entity from a longer text, ideally blending abstractive and extractive methods, a task that has received relatively little attention [5], [6]. Entity-specific summarization is sometimes referred to in literature as a prioritization task,

[1]Disclaimer: The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Any references to commercial entities, products, services, or other nongovernmental organizations or individuals are provided solely for the information of individuals utilizing this material. These references are not intended to reflect the opinion of the Air Force, Department of Defense, the United States, or its employees and may not be quoted or reproduced for the purpose of stating or implying endorsement or approval of any product, person, or service. This information reviewed and cleared for public release by LeMay Center/PA on 18 Dec 23; Case number AETC-2024-0135

[2]The pull request, data, and details can be found at https://huggingface.co/datasets/bloomberg/entsum/discussions/1

where the model must decide which triples from an existing knowledge graph are most representative for a given entity [7] [8], which is distinct from the way we use the term.

Past work has shown that entity-specific summaries are sufficiently distinct from general summaries to require their own methods [5], however, the idea of constraining summary output has been explored in multiple works. Work in constrained summarization introduced the idea of creating entity chains from entities found in the original text to ground summarized output and showed that altering the entity chain could produce altered summaries [9]. However, this work focused on chains and summaries representing entire documents [10], [11] or used tokens other than entities to ground the summary [12].

A substantial difference between this work and past efforts is that previous summarization research has frequently relied on training or fine-tuning a custom model [4]–[6], [9]–[12]. Our experimentation primarily comes from exploring different prompting and generation strategies with LLMs to achieve state-of-the-art results with no fine-tuning. While this work was ongoing, another study of the efficacy of LLMs for summarization was completed, however, their study was small in scale and focused on entire documents [13]. To our knowledge this is the first work to attempt to leverage LLMs for the task of entity-specific summarization.

The summary quality metrics used in this paper are drawn from a number of sources [1], [5], [14] and are discussed in greater detail in Section VI below. While none of the metrics used here are novel, the combination of metrics/measures of quality and how to apply them proactively for a responsible deployment are, to our knowledge, unique.

A. Problem and Approach

The problem primarily discussed in this text is twofold: can an LLM summarize the content of a given context relevant to only a single entity of choice, combining the benefits of extractive and abstractive summarization with minimal or no fine-tuning; and how might we evaluate the text generated by the model to measure its performance?

In this paper we use the entSUM benchmark dataset and apply different models, generation strategies, and prompting strategies to achieve the most promising results while providing an applied survey of the strategies on the task of entity-specific summarization. Results are measured by the metrics originally used in the EntSUM paper [5] as well as with additional metrics that could be used to determine model performance and viability in a production environment, an improvement over how summary model performance is typically reported. These metrics are used to produce a framework for programmatic identification of responses that are likely not suitable for any end user.

The rest of this paper will be structured as follows: Data and Model Selection, Prompting Strategies, and Generation Strategies are explained in Sections II through V, respectively. Section VI explains the criteria used to evaluate results while Section VII contains the results themselves. Section VII presents a discussion of the implications of our results. Finally, Section IX provides concluding thoughts. Additional discussion of data issues and the full prompts used are outlined in Appendixes A and B, respectively. Appendix C contains additional discussion of selected results.

III. Data and Model

A. Data

The EntSUM dataset was used as the benchmark dataset for this effort [5]. EntSUM contains 3,655 entity-specific annotations, 867 of which are second annotations for an already-annotated entity/document pair, from documents that were originally part of the Annotated New York Times Corpus [5], [15]. The dataset follows Kedzie et al. in using articles from 2006 as test data [16].

There are myriad issues with the EntSUM dataset as posted on huggingface [17]. For instructions on how to download the dataset without requiring additional cleaning as well as a detailed list of data issues, see Appendix A. As part of this paper, we publicly release a cleaned version of the the dataset via huggingface that addresses some issues [17]. In the cleaned data, issues involving punctuation at the beginning of an entity's name were corrected. Any row with a "true" summary that was blank was deleted. Other issues outlined in Appendix A were left unaddressed and are an opportunity for future work. Despite its flaws, entSUM still appears to be the best open-sourced dataset to use for entity-centric summarization tasks.

B. Model Selection

Multiple models were selected for exploration. One of the primary drivers of which models to choose was the license the model was released under. A well known family of models was excluded entirely because its license precludes use by the organizations involved in this work. Parameter size was also a primary constraint, due to the hardware limitations outlined below. Ultimately two different model architectures and multiple models within those architectures were included in experiments for this work.

1) Falcon: Due to their open license, the Falcon family of models was identified as the primary model to explore in this effort. The Falcon family of models are decoder-only models [18] trained primarily on the RefinedWeb dataset [19]. Both the 7 billion parameter and 40 billion parameter model were initially identified as potential models, however, results with the 7 billion parameter model were so poor that they are not reported in detail [18]. While this work was ongoing, a 180 billion parameter version of Falcon was released, however, it was too large to run on the hardware available [20].

2) GPT: GPT models are also decoder-only models, though little is known about their training and architecture since they are closed-source and the technical reports omit details [21], [22]. Exact parameter counts are unknown but have been reported to be as high as 1.7 trillion for GPT-4 [23]. Despite their opacity, OpenAI's latest models represent the upper limit of LLM capability [22].

3) Hardware Constraints: The vast majority of this work, with the exception of GPT-3.5 Turbo, 4, and 4 Turbo, was run on the MIT-Lincoln Lab Supercloud, a high performance computing cluster [24]. Experiments were performed on Intel Xeon Gold 6248 nodes with 40 cores, 384GB of RAM, and two Nvidia V100 GPUs each with 32GB ram for a total of 64GB GPU RAM.

4) Quantization: Due to hardware constraints, all runs using Falcon (on Supercloud) were run quantized to 4-bit precision. Recent work indicates that quantization can reduce memory footprint and inference time while sacrificing little in performance [25]. The specific implementation of this quantization was pulled directly from huggingface documentation [17].

IV. Prompting Strategies

The exact phrasing of the prompt given to an LLM plays a large, if unpredictable, role in its performance on a given task. A significant amount of research has been vectored towards refining the prompting strategies used to elicit responses from LLMs, ranging from surveys of style [26], explorations of specific methods [27], "tuning" prompts using traditional ML workflows but without changing model weights [28], to using LLMs themselves to improve their prompts [29], [30]. While the conversational nature of their inputs and outputs makes it tempting to anthropomorphise LLMs, it is important to note that they do not necessarily respond to changes in input in the same way a human would, which adds to the difficulty of manually tuning prompts [31]. Multiple prompting strategies were explored in this research, each of which is outlined below. Each is cited as it appeared in literature originally, but the online survey of techniques from [32] provided a hub from which the literature could easily be found. A list of all the prompts used in runs whose results are reported can be found in Appendix B.

A. Zero-Shot

The initial hypothesis of this project was that LLMs would be able to perform entity-centric summarization effectively in a zero-shot manner. Instruct-tuned models have proven effective at zero-shot tasks after being trained on many possible tasks [33]. For the zero-shot prompt, the model is told to create a summary for a specific entity given the context and given parameters regarding length.

B. Extractive-Abstractive

The authors of the EntSUM paper later reframed entity-specific summarization as a sentence selection task [6]. However, for the reasons outlined above, a purely extractive approach is not always desired. A hybrid approach that first determines relevant passages (Extractive) and then passes the relevant passages to a summarization model (Abstractive) may achieve the best of both worlds, as highlighted in past work [34], [35]. This pipeline is similar to the workflow proposed by Baumel but substitutes a NER-driven relevance approach for their seq2seq-driven relevance model [34]. Sharma used entity clusters to determine sentence relevancy for summarization but focused on top-k or entities found in the leading three sentences, our relevance approach is similar to theirs if only a single entity cluster was selected by a user [11]. For this prompting strategy, Named Entity Recognition (NER) using the spaCy package's transformer model (en_core_web_trf) is performed on the target document to extract all sentences that contain the candidate entity or any individual word found in the candidate entity (e.g. if the candidate entity is "John Smith" any sentences containing "John," "Smith," or "John Smith" would be included). [36]. These sentences are compiled into a new document and only this text is passed to the LLM, which is then asked to summarize the response. In contrast to the other prompting strategies, this prompt instructs the model to create a generic summary because the document has already been filtered to contain only sentences that are, ideally, relevant to the entity of interest. The results of only the Extractive portion of this prompting strategy are reported as the "NER" model.

C. Question-Answer

Instead of asking the LLM to summarize the context in a way that focuses on an entity, the Question-Answer approach simply asks the model to answer "Who is this entity?" using only the context provided.

D. Chain of Thought

The Chain of Thought prompt is very similar to the zero-shot prompt but includes the words "Let's think step by step" along with some additional instructions. Though a seemingly minor adjustment, past work has shown that encouraging an LLM to break a complex task down into tangible steps can greatly improve model performance [27].

E. CESL

The CESL approach could be described as a combination between a few-shot and persona prompt, borrowing elements of our extractive-abstractive approach [26]. Past work suggests that when provided with examples, a generic, instruct-tuned LLM can achieve outstanding performance, though work has primarily focused on classification and multiple choice tasks, not generation of prose [21], [37]. In contrast to the extractive-abstractive approach, which used an additional NER step to extract relevant sentences before summarizing them, CESL simply instructs the model to extract all sentences relevant to an entity and summarize them if they exceed three sentences. Ideally, this produces the exact text found in the original document where appropriate and a lightly edited summary if too many relevant sentences were found, combining elements of traditional abstractive and extractive summarization. Due to input token limitations of the models used, producing multiple full examples of an entity-specific summary was not possible. As a result, "dummy" examples such as the below were used:

Entity: Alice

Context: Alice went to the park. Bob went to the store. Then Alice and Bob went to a movie. Chris went to the zoo.

Edited Context: Alice went to the park. Then Alice and Bob went to a movie.

There was a massive difference in performance and model behavior when given this prompt between Falcon 40b and GPT-4 Turbo, discussed in detail below. To attempt to rectify some of the issues, a more detailed set of examples that looked closer to the training data was given to Falcon 40b and is referred to as the "Improved Few-Shot" prompt in Tables I, II, and III.

V. Generation Strategies

Like any deep learning model, LLMs have parameters that can be adjusted both during training and at inference time. The combination of various parameters adjusted at inference time are referred to here as generation strategies. Multiple generation strategies were tested during this work to explore their impact on results. Implementations for each strategy were taken from huggingface documentation [17].

A. Multinomial Sampling

A multinomial sampling approach treats the selection of each additional token as a probabilistic event and samples the next token based on their probabilities. The parameter 'temperature' increases the weight of less-likely tokens as it increases i.e. a higher temperature will lead to more diverse responses by the LLM.

B. Greedy Decoding

A greedy decoding generation strategy takes the most likely token at each step. It is the fastest of the strategies used in this paper and in contrast to a sampling approach, is deterministic.

C. Beam Search

Beam search explores multiple possible result paths (i.e. beams) to ensure that the most likely overall string is passed back to the user. This approach produced promising results but for long documents, took too much VRAM for the hardware available. Ultimately a beam search generation strategy was not used for any reported results.

VI. Evaluation Criteria

While there are a plethora of available metrics for evaluating summaries [1], there is no single metric that is most suitable for evaluating performance, especially when the summary is intentionally not representative of the entire original text. To address the shortcomings of individual metrics, we use a combination of lexical, semantic, entity-centric, and fact-centric metrics.

A. Lexical Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [38] is a commonly used metric for summarization tasks that relies on finding matching words or sequences of words in the predicted and ground truth summaries. There are multiple variants of ROUGE. ROUGE-N is an n-gram recall between a candidate summary and a reference summary, while ROUGE-L compares the longest common subsequence found between the candidate and reference [38]. Especially using variants of longer N or L, it correlates well with human evaluations of summaries [1]. For this paper, we rely primarily on ROUGE-N with N of 1 and 2, as well as ROUGE-L F1 scores.

B. Semantic Metrics

While ROUGE measures lexical similarity, an obvious limitation of exact word matching is that a text could be extremely similar in meaning (i.e. semantically similar) with minimal lexical overlap. For this reason, we use BERTscore in addition to ROUGE metrics to determine summary quality [39]. BERTscore computes a similarity score for each token in the candidate sentence with each token in the reference sentence. However, instead of exact matches, it compute token similarity using contextual embeddings [39].

C. Entity-Centric Metrics

Previous work has noted that abstractive models can achieve a higher ROUGE score but may contain hallucinated or incorrect information [40], [41]. How these hallucinations should be quantified is a less well-defined question [14]. To evaluate the generated summaries in this project, we use the reference texts created with the method for extractive-abstractive summarization outlined in Section IV.C. This approach is inspired by the work of Hofman-Coyle and Raghavan, who defined entity-specific summarization as a sentence selection task and defined a "Type 1" hallucination, respectively [6], [14]. Any entity found in the model output but not in the reference text is considered a "hallucinated entity". An entity found in the reference text but not in the model output is considered a "missed entity." An entity that is found in both the reference text and the model output is considered a "matching entity." The input text is used as the "gold standard" for this measure for two reasons: it would be possible for the model to include an entity that the annotated summary did not without being incorrect, and there would not be an annotated summary available in a production setting. This is a harsh measure, since there is no fuzzy-matching computed e.g. if a summary contains the entity "American" but the original text said "America," the entities would not match and would be counted as both a hallucinated entity, because "American" is present in the summary, and a missed entity, because "America" did not appear in the summary. It is also important to note that while considered a hallucination by this measure, not all information extraneously included by the model is factually incorrect. In several cases, correct biographical information was included in a response that was not in the context provided. See Section VIII.D below for more on this topic.

D. Fact-based Metrics

Lexical, semantic, and entity-centric metrics together are strong, but still lack the ability to directly compare the

factual consistency of generated responses. Factual accuracy of a summary is a difficult determination to make, though several recent works have taken steps to do so either directly [42], [43], by using Entity F1 score as a proxy for factual consistency [12], or by linking summaries to a knowledge base [41].

To assess the factual consistency of generated summaries, we create a heuristic measure of the proportion of ”correct” sentences in a summary. This proportion is calculated using a modernized python implementation [44] of a package called FactCC [43]. FactCC uses a BERT-based model that takes as input two sentences and assesses factual consistency between them. To determine the proportion of correct sentences, each sentence in a summary is compared to every sentence of the representative text used to measure entity-centric metrics. If the representative text contains a sentence that would make the candidate sentence in the summary ”correct,” that sentence gets a score of 1. If not, it receives a score of 0. If a generated summary has four sentences and three of them return correct and one incorrect, the summary would receive a score of 0.75.

E. Duplicate Handling

Where there are duplicate summaries for entity/document pairs, we depart slightly from how the EntSUM authors handled scoring. While they took the highest ROUGE or BERTscore values, potentially computing scores based on two different summaries, we generate two summaries, compare them to their respective label, and take the higher scoring of the two observations [5]. Since multiple metrics are involved, ”highest scoring” is calculated by averaging ROUGE-L F1 and BERTscore F1. There is substantially more range found in ROUGE-L (typically approximately 0-1 as opposed to 0.8-1 for BERTscore) meaning that ROUGE-L is the primary factor. Where the same entity name was present as both a PER-type entity and an ORG-type entity, results were deduplicated only within the same type. After addressing duplicate summaries and dismissing empty truth summaries, the final number of summaries used for evaluating experiments is 2762.

VII. Results

Table I contains a summary of the results by the metrics used in the entSUM paper, including nine different combinations of prompting and generation strategies run through Falcon 40b, the strictly extractive NER method, and two prompting strategies run with GPT-4 Turbo. Four results are selected from entSUM to be reported here as benchmarks: $\text{Lead3}_{\text{ent}}$, a heuristic that takes the first three sentences that mention an entity as the ”summary” for that entity; $\text{BERTSum}_{\text{ent-top3}}$, the highest performing extractive method from entSUM; $\text{GSum}_{\text{ent-sent}}$, the highest performing abstractive method from entSUM; and Oracle $\text{Lead3}_{\text{ent-summary}}$, which uses all of the sentences manually tagged by the annotators as having been used to create their summary.

In this direct comparison, results from the GPT-4 Turbo/CESL experiment surpass results from entSUM [5] by every metric, and critically, outperform the $\text{Lead3}_{\text{ent}}$ heuristic, a bar that the original results could not clear. This experiment is marked as “Ours.” The CESL experiment also produced summaries with the shortest average length of any method attempted in this paper or in the entSUM paper. Performance in ROUGE metrics approaches that of the Oracle $\text{Lead3}_{\text{ent-summary}}$ and exceeds the Oracle $\text{Lead3}_{\text{ent-summary}}$ in BERTscore. CESL is the only experiment that was able to beat the the highest performing results from [5] in ROUGE, however, all LLM experiments were able to improve on the BERTscore results found in the original paper. Results of the NER method are strong, and though not reported, are achievable an order of magnitude faster than LLM-generated results. The Falcon 40b results are the weakest, with variation based on the prompting and generation strategies discussed further in the following Section.

Table II contains additional metrics of interest that should be considered when determining the best summary model for an organization to use, including generation time; hallucinated, missed, and matching entities; and the mean proportion of ”factually correct” sentences. There is no comparison to the reported entSUM results here because these metrics were not computed in the original paper. For entity metrics, median is used instead of mean because the distributions are highly right skewed. The NER method is excluded from Table II because it is the reference text that the metrics are calculated from. Again, GPT-4 Turbo achieves the best results in most metrics, though the extractive-abstractive Falcon experiments are not far behind and actually outperform GPT-4 Turbo in Median Missed Entities. The extractive-abstractive Falcon experiments were the strongest of the Falcon runs by every metric, which is not surprising given their performance in ROUGE and BERTscore metrics from Table I. Generation time is included in Table II but is not an entirely fair metric to compare between GPT-4 Turbo and other experiments, as they were not running on the same hardware. However, a discussion of the impact of generation time is valid for those who may have similar hardware and are looking to deploy AI systems to production. For more discussion of evaluation criteria and deploying an LLM-driven summary tool to production, see Section VIII.D.

Table III shows the difference in performance for multiple models and prompting strategies for entities that do or do not have corresponding Wikipedia pages. This method for comparing results was driven by early observations that certain responses seemed to be factually correct but contained information not present in the context provided. As skimming the entirety of each model’s training data is impossible, the presence or absence of a Wikipedia page is used as a proxy for an entity who may or may not have been well represented in training data. Notably, there was a large discrepancy in the performance of Falcon 40-b between the two groups. This table and the interpretations of the results are discussed in detail in Section VIII.C.

Figures 1 and 2 show the ROUGE-1 F1 score of two different experiments against a number of ”flags” found for a given observation. A flag is a potential tipper that a generated summary may be of low quality, such as ”Is the generated summary longer than the original text?” The process used to generate the flags and plots as well as details of the implications are found in Section VIII.D.

TABLE I
RESULTS OF MODEL RUNS AGAINST ROUGE AND BERTSCORE METRICS. RESULTS IN **BOLD** BELONG TO THE STRONGEST PERFORMING MODEL FOR A METRIC, RESULTS IN *italics* INDICATE PERFORMANCE BETTER THAN PREVIOUSLY REPORTED NON-ORACLE RESULTS. RESULTS FROM ENTSUM ARE TAKEN DIRECTLY FROM THE ORIGINAL PAPER [5].

Baseline Model	Prompt Strategy	Generation Strategy	ROUGE-1	ROUGE-2	ROUGE-L	BERTscore	Mean Sent./Word
Falcon 40-b	Zero-Shot	Greedy Decoding	31.01	15.51	22.27	*86.60*	7.44 / 176.01
	Chain of Thought	Multinomial Sampling$_{temp=0.2}$	33.61	21.66	26.16	*86.61*	11.30 / 223.83
	Chain of Thought	Greedy Decoding	33.15	21.08	25.65	*86.50*	11.37 / 224.46
	CESL (Ours)	Multinomial Sampling$_{temp=0.2}$	25.52	12.91	19.17	*83.78*	14.96 / 196.39
	Improved Few-Shot	Multinomial Sampling$_{temp=0.2}$	31.10	18.19	22.57	*85.53*	9.92 / 209.81
	QA	Multinomial Sampling$_{temp=0.2}$	33.09	15.20	23.13	*85.51*	7.04 / 132.71
	Extractive-Abstractive	Greedy Decoding	55.15	43.30	47.50	*90.67*	4.47 / 100.40
	Extractive-Abstractive	Multinomial Sampling$_{temp=0.2}$	55.50	43.65	47.86	*90.72*	4.46 / 101.03
	Extractive-Abstractive	Multinomial Sampling$_{temp=0.5}$	51.42	37.75	43.02	*89.95*	4.57 / 100.07
NER	None	NER	62.74	58.31	62.74	*92.96*	6.55 / 188.21
GPT-4 Turbo	CESL (Ours)	Multinomial Sampling$_{temp=0.1}$	**74.03**	**67.26**	**68.43**	**94.40**	2.74 / 78.23
	QA	Multinomial Sampling$_{temp=0.1}$	45.87	27.92	33.92	*88.34*	5.51 / 126.84
Reported EntSUM Results	NA	Lead3$_{ent}$	68.41	60.51	65.03	80.08	2.76 / 92.31
	NA	BERTSum$_{ent\text{-}top3}$	67.8	59.7	64.4	77.89	2.49 / 81.53
	NA	GSum$_{ent\text{-}sent}$	61.45	52.04	58.37	75.87	3.33 / 99.62
	NA	Oracle Lead3$_{ent\text{-}summary}$	85.22	80.49	82.21	91.48	2.53 / 86.0

TABLE II
MODEL RUNS ALONG WITH THEIR GENERATION TIME AND ENTITY/FACT-CENTRIC MEASURES OF SUCCESS. RESULTS IN **BOLD** REPRESENT THE STRONGEST PERFORMING EXPERIMENT FOR A METRIC.

Baseline Model	Prompt Strategy	Generation Strategy	Mean Gen Time (s)	Median Hallucinated Entities	Median Missed Entities	Median Matching Entities	Mean Prop. Correct Sents.
Falcon 40-b	Zero-Shot	Greedy Decoding	82.12	6	8	3	0.689
	Chain of Thought	Multinomial Sampling$_{temp=0.2}$	111.32	11	7	4	0.567
	Chain of Thought	Greedy Decoding	111.12	11	7	4	0.572
	CESL (Ours)	Multinomial Sampling$_{temp=0.2}$	104.30	10	9	2	0.685
	Improved Few-Shot	Multinomial Sampling$_{temp=0.2}$	119.88	8	7	4	0.618
	QA	Multinomial Sampling$_{temp=0.2}$	63.94	8	9	2	0.611
	Extractive-Abstractive	Greedy Decoding	25.38	1	**3**	5	0.906
	Extractive-Abstractive	Multinomial Sampling$_{temp=0.2}$	25.67	1	**3**	5	**0.910**
	Extractive-Abstractive	Multinomial Sampling$_{temp=0.5}$	25.28	1	5	5	0.860
GPT-4 Turbo	CESL (Ours)	Multinomial Sampling$_{temp=0.1}$	4.11	**0**	4	**7**	**0.910**
	QA	Multinomial Sampling$_{temp=0.1}$	5.70	2	6	5	0.684

Overall, the answer to our initial question (Can an LLM summarize the content of a given context relevant to only a single entity of choice with minimal or no fine-tuning?) is yes- if you have the infrastructure and capital to use one of the most powerful LLMs available to the public. GPT-4 Turbo outperformed the previous state-of-the-art models in this task by every metric, producing cohesive and high quality summaries. The answer to our second question (How might we evaluate the text generated by the model to measure its performance?) has been discussed previously in Section VI and is analyzed in more detail below.

VIII. DISCUSSION

The rest of the discussion proceeds as follows: Subsections A and B discuss the impact of model architecture and prompting/generation strategies, respectively. Subsection C focuses on an interesting but incidental finding: the inability of LLMs to disregard learned information about an entity outside of a provided context. Subsection D discusses how our evaluation criteria and findings might be used to deploy an LLM-driven generation tool to production.

Though they did not achieve the best results, substantial time is spent discussing the performance of the Falcon family of models. Why? From an academic perspective, GPT-4 Turbo has achieved state-of-the-art performance and answered our questions. However, from a practitioner's perspective, GPT-4 Turbo is closed source, can only be accessed by sending data to OpenAI via a paid API, and presumably requires massive compute power to run even if it could be run locally, meaning that for organizations seeking to use LLMs on private data holdings or behind a firewall, GPT-4 Turbo is likely not a feasible solution. Just as BERT [3] remains a tool of choice for NER, smaller, open-source LLMs like Falcon are likely to remain tools of choice for tasks where they can be relied upon to perform well until the hardware required to run models like GPT-4 becomes commoditized.

A. Effect of Model Architecture on Performance

The difference in performance between GPT-4 Turbo and Falcon was substantial. It is not a surprise that GPT-4 Turbo outperformed a 40 billion parameter Falcon model, as GPT-4 is reported to have 1.7 trillion parameters [23] and achieved state-of-the-art results in its technical report [22]. Determining any underlying causes in this difference in performance beyond parameter size is difficult, as little is currently known about the inner workings, training processes, and training data

of GPT-4 Turbo. There are still significant advantages of using Falcon, including the fact that it is open-sourced and can be run locally. A more in depth study of the effect of model architecture on entity-specific summary performance would require involving additional models.

B. Effect of Prompting Strategy on Performance

The prompt strategy used for an experiment had a large role in its results. This is most notable in the GPT-4 Turbo experiments, where asking a similar question in a slightly different way and adding some token examples results in a 28.16 point boost in ROUGE-1 score. The phrasing of a prompt also impacts the length of the generated text, and therefore the generation time for each summary. Prompt tuning techniques noted in Section IV could likely drive results even higher, but are left as an opportunity for future work in large part because there is not a suitable training dataset to tune on.

1) Few-Shot Prompt Performance: Results for few-shot prompts were unexpectedly poor for Falcon models. While previous work [21] has noted successes of few-shot prompts, Falcon 40b struggled and frequently returned the few-shot examples in its generated response when provided with simplistic examples- the beginning of the first prompt ("Entity: Alice") was found in 30.2% of its answers. As shown in Table I, the overall performance of Falcon-40b with a few-shot prompt scheme was poor, achieving the worst results by all but one metric (Hallucinated Entities, where it had 10 instead of the worst, 11). When the few-shot examples were more complex, Falcon performed better in terms of both ROUGE (+5.58 ROUGE-1) and BERTscore (+1.75) and returned the examples in its answer far less often. The example entity and leading token "Context: John Smith" was found in 0.14% of responses. However, the structure of the examples (a distinct "//" character) was found in 40.3% of responses, as the model tended to keep providing summaries for additional entities it was not prompted for. In contrast, GPT-4 Turbo made extremely effective use of the simple few-shot examples, producing the strongest results of any experiment. This presents a substantial opportunity for future work. Open questions include: how close to the real data few shot examples must be to increase performance; the impact of model size, structure, and generation settings on few-shot effectiveness; and which tasks can see performance gains with few-shot prompting.

2) Extractive-Abstractive Performance: The extractive-abstractive prompting strategy achieved strong results according to our metrics, outperforming reported entSUM results in BERTscore, though it failed to outperform the previous state-of-the-art ROUGE baselines established in the entSUM paper [5]. However, the more apt comparison is to the NER experiment, which produced the input context of the Extractive-Abstractive experiment. Examined thus, adding a Falcon LLM layer to the pipeline actually reduced performance by BERTscore and ROUGE metrics. This is potentially a failure of the metrics used for the task- neither BERTscore nor ROUGE measures the cohesiveness or logical flow of the response. The average summary length when adding the LLM layer did drop substantially- from a mean of 188.2 to 100 words, with some variation based on the generation strategy. The approach taken with the NER method was simplistic and there is ample opportunity for future work.

C. Use of Parametric vs Retrieved Knowledge

The generated response in certain experiments included factually correct information that was not present in the context provided, despite explicit instructions in the prompt to only rely on information in the context. This is consistent with previous work that has shown that if information about an entity is present in a model's training data, performance in retrieval augmented generation (RAG) tasks may be affected [45]. To explore the effect that the model's parametric knowledge of an entity had on performance of these targeted summaries, the presence or absence of an existing Wikipedia page for a given entity was used to divide summaries into two groups [46]. This is intended as a rough heuristic to divide entities into those who likely were better represented in training data and those who were not. As shown in Table III, when Falcon was prompted with a QA-style prompt, the entities who did not have existing Wikipedia entries resulted in summaries with higher mean ROUGE score (+8.98), higher BERTscore (+3.53), and fewer median hallucinated entities (-4) than the entities who did. This suggests that despite being instructed to do so, Falcon 40b is not able to refrain from using knowledge about entities from its training data to answer questions about them. The difference in results given the same prompt against GPT-4 Turbo was consistent with this trend and also statistically significant, however, the effect was much smaller. The difference in mean ROUGE-1 F1 between entities who did not and did have a Wikipedia page was +1.92. Detailed discussion on the apparent discrepancy in GPT-4 Turbo median entities is located in Appendix C; it is found to be insignificant.

Min et al.[37] suggested that an instruct-tuned model may ignore the task defined by in context examples and instructions to instead use a prior from pretraining, a finding our results support. This has implications for the efficacy of RAG pipelines, a promising way to improve factual consistency and give LLMs access to up-to-date information without expensive fine-tuning [47]. Previous work has determined that RAG systems may produce unexpected results if asked a question that has been frequently seen in training data [45] or if conflicts occur between retrieved information and parametric memory [48]. Our results indicate that this trend holds for the task of entity-specific question and answering. In other words, an organization with private document holdings that may offer non-public facts about otherwise well-known entities cannot trust that only their private information would be used in a RAG approach. An additional implication is that a model trained on private data cannot be trusted to generate answers containing only less restrictive retrieved information, even if explicitly instructed to do so.

This result does not apply to other prompting strategies, indicating that the phrasing of the prompt plays a role in the observed effect. The only LLM experiment other

than the QA prompts that sees a statistically significant difference between entities who do/do not have Wikipedia entries is the Extractive-Abstractive prompt, which actually sees the opposite effect. Interestingly, this effect is also seen in the NER results, which serve as the context provided to the Extractive-Abstractive prompt. The magnitude of difference (-3.5 for ROUGE and -0.9 for BERTscore) is nearly identical for the NER and Extractive-Abstractive experiments. Determining the reason that such a divide exists in a methodology that should be more impartial remains an opportunity for future work. It is possible that NER algorithms are more effective at identifying better known entities and therefore an NER driven pipeline works better for well-known names; alternatively, the difference could be explained by the number of tokens found in entity names.

A complicating factor is whether or not the article being summarized was present in RefinedWeb [19] or in GPT models' training data. Limited testing with a smaller 7-billion parameter Falcon model indicates (through returning of URLs and dates not part of model inputs) some or all of the articles used to create EntSUM may have been part of RefinedWeb; confirming this and determining the implications are left for future work.

D. Responsible Production Considerations

CESL is able beat previously reported metrics in the task of entity-centric sumarization, however, that does not mean that it produces flawless summaries. Even our method will produce some output summaries of low quality or containing erroneous information, requiring a human review of any results generated before further use. For this reason, an automated, quantitative method for parsing out low-quality responses at inference time or immediately afterward is required if an LLM-driven system is to be useful in practice. The most immediate issue preventing this is that referenceless summary metrics frequently assume that the summary should be covering the entire input text, which is not true in the case of an entity or topic specific summary [1]. We propose the creation of reference summaries using the NER method outlined above. By extracting all sentences with references to a given entity and treating the extracted text as the reference summary, metrics like ROUGE, BERTscore, compression ratio, and hallucinated entities can be calculated while targeted fact checking can be performed using libraries like FactCC [43]. While performance on these generated reference summaries does not perfectly align with performance on human annotations, there is a loose correlation that can be exploited by combining multiple weak predictors. Combining multiple quality metrics into an ensemble assessment of a summary's quality could provide a somewhat nuanced picture to a user immediately before they decide to use or not use the summary. To test this theory, we created five tests that could create "red flags" for end users. Flags are applied to summaries that:

- Contain a "hallucinated" entity not found in the reference text
- scored under the median score of ROUGE-1 F1 when comparing all summaries to their references

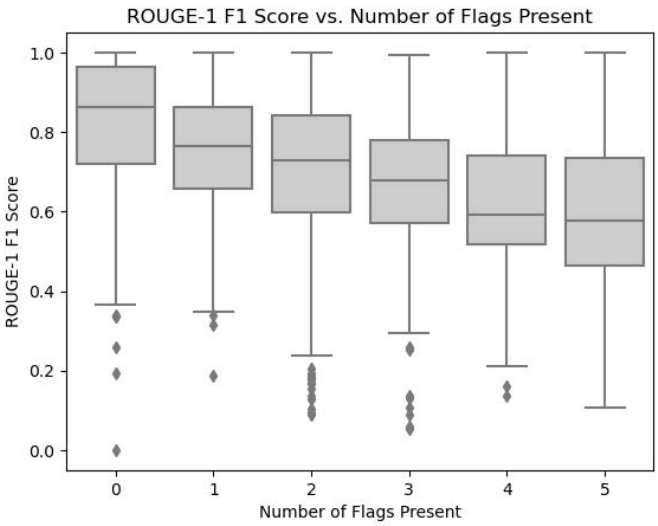

Fig. 1. The number of flags present for a given generated summary is inversely proportional to its ROUGE-1 F1 score against human-created summaries. Critically, no annotation is required to calculate the flags.

- Scored under the median score of BERTscore F1 when comparing all summaries to their references
- Were longer than the reference text
- Had a sentence that did not have a corresponding sentence in the reference text for which FactCC said CORRECT

The results are promising. Figure 1 shows the real ROUGE-1 F1 scores for the highest performance model run against the number of flags present for a response. Responses that had no flags (n=937, 33.9% of the total) achieve a mean ROUGE-1 F1 of 82.28, an 8.25 point jump. As Figure 2 shows, this trend holds even for weaker models, presenting a critical implication for end users: even for an imperfect model, it is possible to automatically identify generated responses that do not meet the mark and separate them from those that might. While the number of observations with no flags decreases for the weaker model, to n=511 from n=937, a 19% reduction in end users work load may still be significant. This reduction does need to be balanced with the time required to review generated summaries to determine if including an LLM-driven system is a net positive for an organization.

There is ample opportunity for future work in evaluating the performance of LLM-generated responses. For example, applying our framework to a RAG system may also help identify problematic responses. Instead of using NER to get candidate sentences, the system could use the retrieved sentences/paragraphs as the candidate summary and present a user with statistics regarding answer quality and potentially problematic tokens/sentences along with the answer.

IX. Conclusion

We show that instruct-tuned LLMs achieve state-of-the-art results using the CESL prompting approach (+6.23 in ROUGE-1 and +16.51 in BERTscore) on the task of entity-specific summarization without additional fine-tuning. We also show that responses to certain prompts regarding well-known entities are significantly affected by a model's

TABLE III
THE DIFFERENCE IN MEAN ROUGE, BERTSCORE, AND MEDIAN HALLUCINATIONS FOR SUMMARIES WHERE THE ENTITY IN QUESTION EITHER HAD OR DID NOT HAVE AN EXISTING WIKIPEDIA PAGE. ALL RUNS ARE THOSE WITH A MULTINOMIAL SAMPLING GENERATION STRATEGY. ROWS ANNOTATED WITH A * INDICATE EXPERIMENTS WHERE THE DIFFERENCE IN MEAN ROUGE-1 AND MEAN BERTSCORE WERE BOTH STATISTICALLY SIGNIFICANT ($P<0.05$)

Baseline Model	Prompt Strategy	Has Wikipedia?	ROUGE-1	BERTScore	Hallucinated Entities
Falcon 40-b	QA*	Yes	29.49	83.64	9
		No	38.48	87.17	5
	CESL	Yes	24.81	83.69	9
		No	24.04	83.27	11
	Improved Few-Shot*	Yes	29.41	84.98	9
		No	31.28	85.55	8
	COT	Yes	32.16	86.33	11
		No	33.85	86.56	11
	Extractive-Abstractive*	Yes	54.63	90.59	1
		No	51.12	89.76	1
NER	None*	Yes	61.81	92.86	0
		No	58.35	91.92	0
GPT-4 Turbo	QA*	Yes	44.47	88.02	2
		No	46.39	88.54	3
	CESL	Yes	71.90	93.98	0
		No	71.74	93.84	0

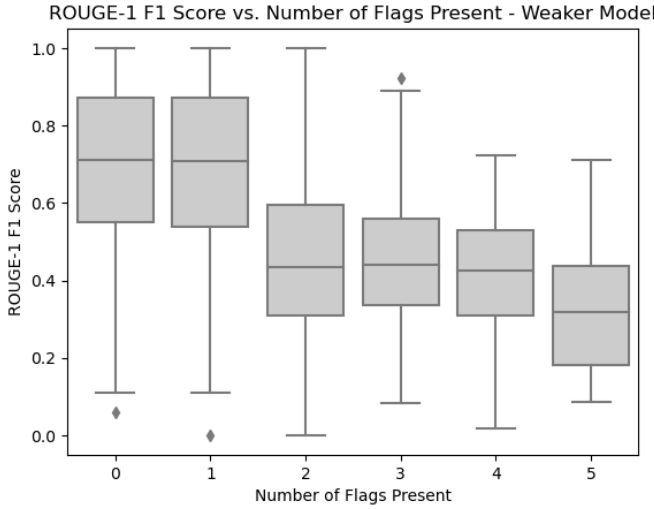

Fig. 2. The inverse trend of flags present to ROUGE-F1 holds, even for a weaker model (Falcon 40b/Extractive-Abstractive/Greedy Decoding). This suggests potential for the methodology to identify useful results from even a limited model.

parametric memory, even if the model is instructed to rely only on retrieved context, a finding that has critical implications for RAG systems and is unavoidable to some capacity in current models. While they achieve state-of-the-art results on entity-specific summarization, there is substantial work yet to be done in validating the quality of outputs from LLMs, for this reason, we recommend using their output as a starting point for manual review and editing. To enable this human-in-the-loop workflow, we offer a framework for quantifying summary model performance and identifying problematic results proactively to focus human reviewers on results that are more likely to be helpful. There are many factors that should be considered before deploying an LLM-driven system to production, such as performance (both absolute and in comparison to non-LLM solutions), inference time, cost, whether the model is closed or open source, level of risk an organization is willing to accept, and most importantly, whether or not the use case is appropriate for a model that may incidentally produce non-factual results. Given the time required for human review, determining benchmarks for when and at what capacity integrating LLMs into workflows becomes a net positive is also a necessity. Lastly we publicly release an updated version of the entSUM dataset to help enable some of this future work. LLMs are an incredible technology that promise vast efficiency gains across many sectors and roles, our findings show that they can even outperform smaller models fine-tuned for specific tasks; however, our results also show why it is important for organizations seeking to deploy LLM-driven solutions to remain clear-eyed about the potential drawbacks of LLMs.

ACKNOWLEDGMENT

The authors would like to acknowledge Laura Niss, Sanjeev Mohindra, Michael Yee, and the rest of the AIA Trustworthy AI team for their contributions and support. We also acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this paper.

REFERENCES

[1] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating summarization evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, B. Roark and A. Nenkova, Eds., pp. 391–409, 2021. DOI: 10.1162/tacl_a_00373. [Online]. Available: https://aclanthology.org/2021.tacl-1.24.

[2] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[4] A. Fan, D. Grangier, and M. Auli, "Controllable abstractive summarization," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 45–54. DOI: 10.18653/v1/W18-2706. [Online]. Available: https://aclanthology.org/W18-2706.

[5] M. Maddela, M. Kulkarni, and D. Preotiuc-Pietro, *Entsum: A data set for entity-centric summarization*, 2022. arXiv: 2204.02213 [cs.CL].

[6] E. Hofmann-Coyle, M. Kulkarni, L. Xie, M. Maddela, and D. Preotiuc-Pietro, "Extractive entity-centric summarization as sentence selection using bi-encoders," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online only: Association for Computational Linguistics, Nov. 2022, pp. 326–333. [Online]. Available: https://aclanthology.org/2022.aacl-short.40.

[7] Q. Liu, G. Cheng, K. Gunaratna, and Y. Qu, "Esbm: An entity summarization benchmark," in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 548–564, ISBN: 978-3-030-49461-2.

[8] L. Chen, Z. Li, W. He, *et al.*, "Entity summarization via exploiting description complementarity and salience," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8297–8309, 2023. DOI: 10.1109/TNNLS.2022.3149047.

[9] S. Narayan, Y. Zhao, J. Maynez, G. Simoes, V. Nikolaev, and R. McDonald, *Planning with learned entity prompts for abstractive summarization*, 2021. arXiv: 2104.07606 [cs.CL].

[10] C. Zheng, Y. Cai, G. Zhang, and Q. Li, "Controllable abstractive sentence summarization with guiding entities," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds., Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5668–5678. DOI: 10.18653/v1/2020.coling-main.497. [Online]. Available: https://aclanthology.org/2020.coling-main.497.

[11] E. Sharma, L. Huang, Z. Hu, and L. Wang, *An entity-driven framework for abstractive summarization*, 2019. arXiv: 1909.02059 [cs.CL].

[12] Y. Mao, X. Ren, H. Ji, and J. Han, *Constrained abstractive summarization: Preserving factual consistency with constrained generation*, 2021. arXiv: 2010.12723 [cs.CL].

[13] L. Basyal and M. Sanghvi, *Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models*, 2023. arXiv: 2310.10449 [cs.CL].

[14] G. Raghavan, *Illusions unraveled: The magic and madness of hallucinations in llms — part 1*, 2023. [Online]. Available: https://www.yurts.ai/post/illusions-unraveled-the-magic-and-madness-of-hallucinations-in-llms-part-1.

[15] E. Sandhaus, *The new york times annotated corpus ldc2008t19*, 2008. DOI: https://doi.org/10.35111/77ba-9x74.

[16] C. Kedzie, K. McKeown, and H. Daumé III, "Content selection in deep learning models of summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1818–1828. DOI: 10.18653/v1/D18-1208. [Online]. Available: https://aclanthology.org/D18-1208.

[17] T. Wolf, L. Debut, V. Sanh, *et al.*, ""transformers: State-of-the-art natural language processing"," in *"Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations"*, 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[18] E. Almazrouei, H. Alobeidli, A. Alshamsi, *et al.*, "Falcon-40B: An open large language model with state-of-the-art performance," 2023.

[19] G. Penedo, Q. Malartic, D. Hesslow, *et al.*, *The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only*, 2023. arXiv: 2306.01116 [cs.CL].

[20] E. Almazrouei, H. Alobeidli, A. Alshamsi, *et al.*, "The falcon series of language models: Towards open frontier models," 2023.

[21] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[22] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[23] R. Albergotti, *Microsoft pushes the boundaries of small ai models with big breakthrough*, 2023. [Online]. Available: https://www.semafor.com/article/11/01/2023/microsoft-pushes-the-boundaries-of-small-ai-models.

[24] A. Reuther, J. Kepner, C. Byun, *et al.*, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, IEEE, 2018, pp. 1–6.

[25] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, 2023. arXiv: 2305.14314 [cs.LG].

[26] J. White, Q. Fu, S. Hays, *et al.*, *A prompt pattern catalog to enhance prompt engineering with chatgpt*, 2023. arXiv: 2302.11382 [cs.SE].

[27] J. Wei, X. Wang, D. Schuurmans, *et al.*, *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL].

[28] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243.

[29] Y. Wang, Y. Kordi, S. Mishra, *et al.*, *Self-instruct: Aligning language models with self-generated instructions*, 2023. arXiv: 2212.10560 [cs.CL].

[30] Y. Zhou, A. I. Muresanu, Z. Han, *et al.*, "Large language models are human-level prompt engineers," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=92gvk82DE-.

[31] A. Webson, A. M. Loo, Q. Yu, and E. Pavlick, *Are language models worse than humans at following prompts? it's complicated*, 2023. arXiv: 2301.07085 [cs.CL].

[32] E. Saravia, "Prompt Engineering Guide," *https://github.com/dair-ai/Prompt-Engineering-Guide*, Dec. 2022.

[33] J. Wei, M. Bosma, V. Y. Zhao, *et al.*, *Finetuned language models are zero-shot learners*, 2022. arXiv: 2109.01652 [cs.CL].

[34] T. Baumel, M. Eyal, and M. Elhadad, *Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models*, 2018. arXiv: 1801.07704 [cs.CL].

[35] H. Li, A. Einolghozati, S. Iyer, *et al.*, "EASE: Extractive-abstractive summarization end-to-end using the information bottleneck principle," in *Proceedings of the Third Workshop on New Frontiers in Summarization*, G. Carenini, J. C. K. Cheung, Y. Dong, F. Liu, and L. Wang, Eds., Online and in Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 85–95. DOI: 10.18653/v1/2021.newsum-1.10. [Online]. Available: https://aclanthology.org/2021.newsum-1.10.

[36] M. Honnibal, I. Montani, S. V. Landeghem, and A. Boyd, "Spacy: Industrial-strength natural language processing in python," 2020. DOI: 10.5281/zenodo.1212303.

[37] S. Min, X. Lyu, A. Holtzman, *et al.*, *Rethinking the role of demonstrations: What makes in-context learning work?* 2022. arXiv: 2202.12837 [cs.CL].

[38] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013.

[39] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: 1904.09675 [cs.CL].

[40] A. Deroy, K. Ghosh, and S. Ghosh, *How ready are pre-trained abstractive models and llms for legal case judgement summarization?* 2023. arXiv: 2306.01248 [cs.CL].

[41] A. Alambo, T. Banerjee, K. Thirunarayan, and M. Raymer, *Entity-driven fact-aware abstractive summarization of biomedical literature*, 2022. arXiv: 2203.15959 [cs.CL].

[42] B. Goodrich, V. Rao, P. J. Liu, and M. Saleh, "Assessing the factual accuracy of generated text," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 166–175, ISBN: 9781450362016. DOI: 10.1145/3292500.3330955. [Online]. Available: https://doi.org/10.1145/3292500.3330955.

[43] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," *arXiv preprint arXiv:1910.12840*, 2019.

[44] M. Deprada, *Modernized factcc implementation*, 2023. [Online]. Available: https://huggingface.co/manueldeprada/FactCC.

[45] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 9802–9822. DOI: 10.18653/v1/2023.acl-long.546. [Online]. Available: https://aclanthology.org/2023.acl-long.546.

[46] Wikipedia contributors, *Wikipedia, the free encyclopedia*, [Online; accessed 8-NOV-2023], 2023. [Online]. Available: https://en.wikipedia.org.

[47] P. Lewis, E. Perez, A. Piktus, *et al.*, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: 2005.11401 [cs.CL].

[48] J. Xie, K. Zhang, J. Chen, R. Lou, and Y. Su, *Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts*, 2023. arXiv: 2305.13300 [cs.CL].

APPENDIX A
USING ENTSUM BENCHMARK DATA

When attempting to use the original EntSUM dataset directly from huggingface [17], the most logical step is to import as a dataset using huggingface's prewritten functions. Downloading the dataset in this way results in messy, duplicated data. The most effective way to use the dataset, at least for benchmarking purposes, is to manually download only the "nyt_annotate_all_annotation" json from huggingface, which returns a cleaner and deduplicated set. Even handled thus, numerous issues remain. While we make no claim to have identified every issue with the EntSUM dataset, identified issues with problematic entries include:

- Entities incorrectly start or end with punctuation like "?"" or "!""
- "True" summaries are blank
- Entity names are incomplete compared to the true text (e.g. entity listed as Kevin P instead of Kevin P O'Donnell, BROWN instead of Michael D. Brown)
- Entries for the same entity/document listed as both an ORG and a PER-type node
- Duplicate entities for the same entity/document with slightly different names (e.g. Mr. Carter and Charles Carter)
- Certain entities have an incorrect type or being incorrectly included at all (e.g. "Blum" tagged as an ORG when it is referencing a person in the text; "S.U.V." is tagged as an ORG-type entity, when it is referencing a classification of cars, not an organization; "Mr. Go Mardi Gras" referencing a costume of a drainage channel worn for Mardi Gras)
- Grammatically incorrect truth summaries (e.g. "In the trial of Louis J. Eppolito, a shy retiring manner, a convicted marijuana dealer took the stand...")
- Quotation marks found after an entity's name (e.g. Emily ")

Additionally, there are many duplicate entities found in the dataset. "Bush," generally referring to former President George W. Bush or his administration, appears 42 times, representing approximately 1.5% of the evaluation data. George Pataki (a former Governor of New York) appears 27 times, approximately 0.98% of the evaluation data. Of 2762 observations used to calculate metrics, there are only 2114 unique entity names. This does not count scenarios noted above where an entity may be referred to by two slightly different names (e.g. Bush and George W. Bush), meaning the true number is lower. We leave fully examining the impact of these duplicated entities as an opportunity for future work.

Finally, in the original paper, the authors claim to have filtered articles used to build the dataset to only those under 1500 words [5]. In an effort to replicate their work, we also filter for only documents under 1500 words, but found that one document (1734268) with a word count over 1500 may have been erroneously included in the data used to calculate the benchmarks for EntSUM. The five entities pulled from this document are excluded from our results. To determine word count, the word_count field from the original NYT dataset is used [15].

APPENDIX B
FULL PROMPTS

This Appendix contains each of the prompts used for reported results.

A. Zero-Shot

Instruction: Write a 2-4 sentence summary of the following context that is focused on parts of the text relevant to entity. Make minor adjustments to the original text to produce a legible, coherent, and helpful response. Use 350 words or fewer. Do not use the words "The article" or "the context" in your response, only the summarized text.
Context: context
Summary:

B. Few Shot

You are a helpful assistant that follows instructions exactly.
Instruction: Write a concise summary of the context focused only on the entity provided. Do not return any of the examples in your response.
Examples
Context: John Smith, a Virginia-based researcher, created a new prototype. Smith worked with Alice Brown, a Maryland-based engineer, to write a new paper about the prototype. Smith met with the Council for Science to discuss their results. Many people are interested in the paper.
Focused Summary: John Smith // John Smith, a Virginia-based researcher, created a new prototype. Smith worked with Alice Brown, a Maryland-based engineer, to write a paper. Smith met with the Council for Science to discuss their results
Context: John Smith, a Virginia-based researcher, created a new prototype. Smith worked with Alice Brown, a Maryland-based engineer, to write a new paper about the prototype. Smith met with the Council for Science to discuss their results. Many people are interested in the paper.
Focused Summary: Alice Brown // Alice Brown, a Maryland-based engineer, worked with John Smith to write a new paper about the prototype Smith created.
Context: John Smith, a Virginia-based researcher, created a new prototype. Smith worked with Alice Brown, a Maryland-based engineer, to write a new paper about the prototype. Smith met with the Council for Science to discuss their results. Many people are interested in the paper.
Focused Summary: Council for Science // John Smith, a researcher, met with the Council for Science to discuss his results.
Response Context: {context} Focused Summary: {entity} //
Note: Results for the following prompt are referenced in the body of the paper but were not reported due to poor performance
System: You are a helpful assistant that follows instructions exactly.

Instruction: Edit the context to contain only sentences mentioning the provided entity. If there are more than three sentences containing the entity, return only the first three. Here are some examples:

Entity: Alice

Context: Alice went to the park. Bob went to the store. Then Alice and Bob went to a movie. Chris went to the zoo.

Edited Context: Alice went to the park. Then Alice and Bob went to a movie.

Entity: Steve

Context: Steve and Tim went to the park. Amanda ate a banana and a grape. Jeremy Brown went to Steve's house. Then Alex, Steven, and Jeremy went shopping. Steve and Jim are friends.

Edited Context: Steve and Tim went to the park. Jeremy Brown went to Steve's house. Then Alex, Steven, and Jeremy went shopping.

Entity: {entity}

Context: {context}

Edited Context:

C. Extractive-Abstractive

System: You are a helpful assistant that follows instructions exactly.

Instruction: Summarize the context into 2-3 sentences. Use as much of the original text as possible.

Context: {context}

Summary:

D. QA Prompt

You are a helpful assistant that follows instructions exactly. You will be given a news article as context and asked a question. Use only the information in the context to answer the question.

Context: {context}

Question: Based only on the context, who is entity? Provide any recent activity, biographical details or relations mentioned in the context

Answer using only the Context:

E. Few-Shot/Extractive

System: You are a text editor. Edit the context to contain only sentences mentioning the provided entity. If there are more than three sentences containing the entity, return only the first three. ### Instruction: You are a text editor. Edit the context to contain only sentences mentioning the provided entity. If there are more than three sentences containing the entity, return only the first three.

Entity: Alice

Context: Alice went to the park. Bob went to the store. Then Alice and Bob went to a movie. Chris went to the zoo.

Edited Context: Alice went to the park. Then Alice and Bob went to a movie.

Entity: Steve

Context: Steve and Tim went to the park. Alice ate a banana and a grape. Jeremy Brown went to Steve's house. Then Alice, Steven, and Jeremy went shopping. Steve and Jim are friends.

Edited Context: Steve and Tim went to the park. Jeremy Brown went to Steve's house. Then Alice, Steven, and Jeremy went shopping.

Entity: {entity}

Context: {context}

Edited Context:

F. Chain of Thought

Instruction: Write a 2-4 sentence summary of the following context that is focused on parts of the text relevant to entity. Make minor adjustments to the original text to produce a legible, coherent, and helpful response. Use 350 words or fewer. Do not use the words "The article" or "the context" in your response, only the summarized text.

Context: {context}

Instruction: Let's think step by step. First determine which sentences are most relevant to {entity}, then summarize only those sentences.

Summary:

APPENDIX C
GPT-4 TURBO MEDIAN ENTITIES

At first glance the difference in median hallucinated entities in the GPT-4 Turbo run appears counter-intuitive, as the entities with Wikipedia pages have a median of 2 and those that do not have a median of 3. Upon further review, the difference is negligible and appears to be a case of edge-case behavior on quantile boundaries. Taking the quantile at 0.494 instead of 0.500 (given n=1084, a difference of 6.5 entities) for entities with no Wikipedia page would give a score of 2 hallucinated entities. Conversely, taking a quantile at 0.522 (n=2529, a difference of 56 entities) would have produced a score of 3 hallucinated entities. Looking at the mean, though generally less robust to skew and outliers, shows the pattern we would expect: the Wikipedia-having entities performing slightly worse.