FanOutQA: Multi-Hop, Multi-Document Question Answering for Large Language Models

Andrew Zhu, Alyssa Hwang, Liam Dugan, Chris Callison-Burch
University of Pennsylvania
{andrz,ahwang16,ldugan,ccb}@seas.upenn.edu

Abstract

One type of question that is commonly found in day-to-day scenarios is "fan-out" questions, complex multi-hop, multi-document reasoning questions that require finding information about a large number of entities. However, there exist few resources to evaluate this type of question-answering capability among large language models. To evaluate complex reasoning in LLMs more fully, we present FanOutQA, a high-quality dataset of fan-out question-answer pairs and human-annotated decompositions with English Wikipedia as the knowledge base. We formulate three benchmark settings across our dataset and benchmark 7 LLMs, including GPT-4, LLaMA 2, Claude-2.1, and Mixtral-8x7B, finding that contemporary models still have room to improve reasoning over inter-document dependencies in a long context. We provide our dataset and open-source tools to run models to encourage evaluation.[1]

1 Introduction

One task that would be particularly useful for LLMs to be able to do is to answer "fan-out" questions: questions that require models to find a list of entities and then consult a large number of documents to aggregate information about those entities to answer a user's question. This pattern of question can be found commonly in day-to-day scenarios, such as performing a literature review (fan-out over research papers), planning a trip (fan-out over attractions), or choosing where to eat (fan-out over nearby restaurants). The fan-out task is particularly challenging because it requires multi-hop reasoning across multiple documents, and the combined length of the documents needed to answer the question typically exceeds the length of a model's context window. Existing question-answering benchmarks like HotpotQA (Yang et al., 2018), LongBench (Bai et al., 2023), and ZeroSCROLLS (Shaham et al., 2023) focus on intra-document dependencies or dependencies between a small number of documents, which does not sufficiently evaluate models' performance on this type of task.

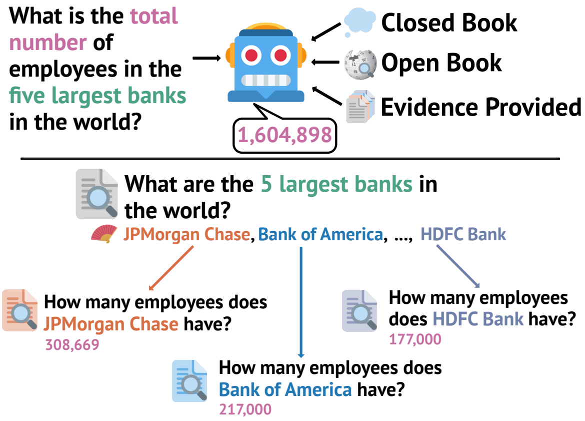
What is the total number of employees in the five largest banks in the world?
Closed Book
Open Book
Evidence Provided
1,604,898
What are the 5 largest banks in the world?
JPMorgan Chase, Bank of America, ..., HDFC Bank
How many employees does JPMorgan Chase have?
308,669
How many employees does HDFC Bank have?
177,000
How many employees does Bank of America have?
217,000

Figure 1: The FanOutQA dataset contains multi-hop, multi-document "fan-out" questions along with human-written decompositions (bottom). We formulate three challenge settings for LLMs to answer these fan-out questions to test capabilities of LLMs (top).

In this paper, we present FanOutQA, a high quality dataset of 1,034 information seeking questions, 7,305 human-written decompositions, and their answers, along with a multi-hop, multi-document benchmark using English Wikipedia as its knowledge base. Compared to other question-answering benchmarks, FanOutQA requires reasoning over a greater number of documents, with its main focus being on the fan-out style of question (Figure 1).

We formulate three distinct challenge settings over the dataset. The **closed-book** setting requires the model to answer fan-out questions without external knowledge, testing its general knowledge. The **open-book** setting gives models access to retrieval tools, testing their ability to retrieve relevant articles and reason across multiple long documents. Finally, the **evidence-provided** setting provides the models with relevant articles, testing their long-

[1] https://fanoutqa.com
https://github.com/zhudotexe/fanoutqa

context and multi-hop reasoning capabilities.

We find that the closed- and open-book settings are difficult for modern systems, with the best performing models scoring below 50%. In the open-book setting, retrieved documents outgrow models' context lengths. In the evidence-provided setting, models' performance correlates strongly with their context length. Human volunteers completing the open-book task score 85% accuracy, showing room to improve LLM systems.

2 Related Work

Multi-Hop Question Answering. HotpotQA (Yang et al., 2018) focuses on using bridge entities to introduce a "hop", requiring models to retrieve information about two related entities. ComplexWebQuestions (Talmor and Berant, 2018) composes simpler questions to create two-hop questions with a similar bridge entity. 2WikiMultiHopQA (Ho et al., 2020) uses manually curated templates to generate two to four-hop questions among entities in the same class. MuSiQue (Trivedi et al., 2022) presents algorithmically generated questions with nonlinear reasoning chains, which require up to four hops per question. These datasets focus on simple reasoning chains, with a maximum of four hops. In FanOutQA, we require nonlinear reasoning chains that are longer than previous multi-hop QA datasets (an average of seven hops per question).

Long Context Evaluations. LongBench (Bai et al., 2023) is a collection of multiple long-context tasks. In its multi-document QA setting, it builds on top of the multi-hop QA benchmarks discussed above, adding distractor spans to create artificial long documents which are provided to the model. However, it has been shown that this approach does not necessarily increase the complexity of the QA task (Min et al., 2019). The Qasper (Dasigi et al., 2021) and SCROLLS (Shaham et al., 2022) benchmarks present QA tasks that focus primarily on reading comprehension within a single document, rather than reasoning across multiple documents. These benchmarks and others also evaluate different aspects of long context reasoning through subjective summarization tasks (Kwan et al., 2023) or text span reordering (Shaham et al., 2023; Li et al., 2023), which is beyond the focus of our benchmark. Unlike previous benchmarks, our open-book setting requires models to *retrieve* and reason over multiple natural long documents (*multi-hop multi-document*), and our evidence-provided setting requires models to perform inter-document reasoning over multiple provided documents. On average, questions in FanOutQA are paired with 172k tokens of evidence spanning 7 documents.

3 FanOutQA Dataset

FanOutQA consists of three parts: questions, answers, and evidence. Each question includes a decomposition into sub-questions that can be answered with a single Wikipedia article. The answers to the sub-questions can then be combined to answer the top-level question. We provide these sub-questions, answers, and associated Wikipedia articles as an additional resource for decomposing complex queries. Sample questions are provided in Appendix A.

3.1 Dataset Creation

To create FanOutQA, we recruited 379 undergraduate and graduate students enrolled in AI or NLP courses at a US university to write questions and answers in the fan-out style. We required each question to reference at least five different Wikipedia articles to find its answer. We also tasked the students to decompose their top-level questions into sub-questions, each providing an answer from a single article. The questions were written in a period of one week, ending on November 20, 2023. We stored a snapshot of Wikipedia on the last day to preserve the knowledge source, which we provide with the dataset. We provided a Jupyter notebook to help with writing (see Appendix E) and offered students extra credit for their contributions.

The students produced 1,418 sets of top-level questions, sub-questions, and Wikipedia references. After our filtering pipeline (Appendix B) to ensure the quality of our dataset, we arrive at 1,034 top-level questions and 7,305 sub-questions, across 4,121 distinct Wikipedia articles. We split the dataset into dev and test splits at a ratio of 30% dev (310), 70% test (724). We release the full questions, decomposition, and answers of the dev questions, and only the top-level question and list of articles used in the decomposition for the test questions. We maintain a leaderboard of performance on the test set on our website, with a standard submission for generations on the test set.

3.2 Settings

We present three different benchmark settings over the data to evaluate different aspects of LLM systems, which we present in order of expected difficulty (most-to-least difficult).

Closed Book. In what could be considered the most difficult setting, the model is given only the top-level question and must answer it based solely on the knowledge encoded in its parameters. This setting primarily tests the model's general knowledge and establishes a model-specific baseline.

Open Book. The open book setting gives the model access to the Wikipedia knowledge base along with the top-level question. Using retrieval tools, it can query our dated snapshot of Wikipedia for relevant information across multiple rounds of interaction. Since the questions in FanOutQA require multiple reasoning steps over specific information across a large number of documents, the open book setting is suitable for evaluating retrieval-augmented generation, multi-hop reasoning, and long-horizon question answering.

Evidence Provided. In this setting, the model is given the top-level question and the text of each Wikipedia article used in the decomposition. The model can answer based on information fully within its context window, which evaluates long-context and long-dependency reasoning similar to Li et al. (2023). It can alternatively retrieve the necessary information from the given documents as a simpler retrieval task.

4 Benchmarking Study

We benchmarked seven large language models on FanOutQA: GPT-4, GPT-4-turbo, GPT-3.5-turbo, LLaMA 2, Mistral-7B, Mixtral-8x7B, and Claude 2 (more details in Appendix C). All models generated text with greedy decoding; all local models were run with FP16 precision.

4.1 Metrics

We report benchmark performance with four classes of metrics.

The first is string accuracy, which we compute after lemmatizing and removing stop words and punctuation from each sequence:

$$Loose(R, g) = \frac{\sum_{r \in R} \mathbb{1}[substr(r, g)]}{|R|} \quad (1)$$

$$Strict(R, g) = \mathbb{1}[Loose(R, g) = 1] \quad (2)$$

Where R is the list of normalized reference answer strings for a given question and g is the normalized candidate generation for that question.

We report the mean proportion of reference answer strings found in the generation ("loose" accuracy, Eqn. 1) and proportion of questions in which *every* answer string was found in the generation ("strict" accuracy, Eqn. 2).

We also report ROUGE-1, ROUGE-2, and ROUGE-L F1-scores (Lin, 2004) and BLEURT (Sellam et al., 2020) scores, consistent with existing related work. Finally, we use GPT-4 (`gpt-4-0613`) to estimate the factual equivalence of the generated and reference answers for each question (prompt in Appendix F). We observe that this method is more robust to misspellings and string substitutions, such as "two" and "2" or "1 trillion" and "1000 billion." We present loose string accuracy and the model judge score across all settings in Figure 2, and tabulate all other results in Appendix D.

4.2 Closed Book Results

Using only knowledge encoded in their parameters, models' loose string accuracy ranged from 0.341 (Claude) to 0.470 (Mixtral), with none reaching our estimated human baseline of 0.685 or upper bound of 0.847 (see Section 4.5).

Most errors were plausible but incorrect hallucinations. For example, when asked "which of the top five best selling video games does not feature physical combat," GPT-4-turbo answered "Minecraft" even though the true answer is Tetris.

A substantial proportion of errors were unique to OpenAI's GPT models. These models often refused to answer, citing lack of real time data. Of the models, GPT-4-turbo refused to answer 5% of the time, GPT-3.5-turbo 10%, and GPT-4 44%.

4.3 Open Book Results

We used Kani (Zhu et al., 2023) to provide access to Wikipedia using native function calling (OpenAI's GPT models) or through a structured search query. We split each retrieved document into 1024-character chunks, preferring to split at paragraph and sentence boundaries. We ranked the chunks with a BM25+ (Lv and Zhai, 2011) retriever and provided up to half the model's context length of tokens per document. Mistral-7B suffered from severe neural text degeneration (Holtzman et al., 2020) and entered infinite loops when attempting to search, so we omit its open-book results.

Perhaps surprisingly, most models performed worse in the open-book setting than in the closed book setting. We find this to be because models in this setting "forgot" the original question as

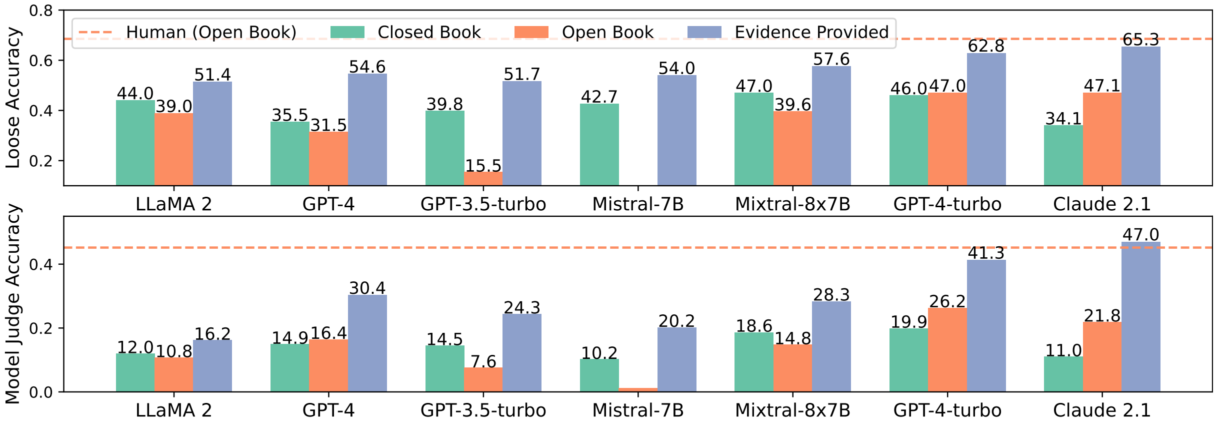

Figure 2: Loose string accuracy and model judged accuracy of all benchmarked models in all settings, including baseline human performance in the open-book setting. See Appendix D for additional metrics.

their context windows filled with long retrieved passages across multiple retrieval rounds, outputting a summary of the last retrieved passage instead of answering the question. This is supported by a moderate positive correlation between maximum context window sizes and model-judged accuracy ($r^2 = 0.558$). Models with larger context lengths are able to include a greater amount of information in the context and "forget" the original question less often as context windows fill up. We ran an additional experiment where we limited the context window of all models to the smallest of all models to verify this correlation, the results of which are tabulated in Appendix D.

4.4 Evidence Provided Results

We use the same retrieval scheme as in the open-book setting, providing models as many chunks as would fit each model's context. Performance correlated strongly with maximum context length in this setting ($r^2 = 0.782$), supporting the proposition that the amount *and quality* of information in a model's context affects its ability to answer fan-out questions. This shows that questions in FanOutQA effectively measure long-context reasoning over very long dependencies.

4.5 Human Performance

We conducted a human evaluation to create a human baseline and estimate the upper bound of human performance on FanOutQA. We recruited 14 volunteers to each answer 10 FanOutQA questions with access to Wikipedia, similar to the open-book setting. On average, humans took 5-15 minutes to answer each question. In the open-book setting, the humans score significantly higher than our tested models ($p < 0.05$), achieving a loose accuracy of 68.5% and model-judged accuracy of 45.2%. This score may seem low, as the model-judged accuracy does not account for partial credit. As our only automated metric that accounts for partial credit is not robust to typos and equivalent string substitutions, we also manually evaluate the human answers to establish an upper bound of 84.7%.

5 Conclusions

Fan-out question answering presents several challenges for LLMs, including decomposing complex questions into simpler sub-questions, retrieving documents, extracting relevant information, and multi-hop reasoning over a large number of documents. We developed a dataset called FanOutQA for this ambitious task in response to the rapidly improving reasoning abilities and context management strategies in large language models, and we formulate three challenge settings over the dataset. We benchmarked the performance of seven state-of-the-art models on our challenge settings, and find that the requirement of fan-out question-answering challenges even the long context capabilities of modern models. Accuracy correlated with context length in the open book and evidence-provided but not in the closed book settings, suggesting that more information helps performance. The correlation was stronger in the evidence-provided setting, further suggesting that the quality of information matters as well. We encourage researchers to use FanOutQA to evaluate new retrieval-augmented models, long-context models, and other novel LLM systems with our open-source resources.[2]

[2] https://fanoutqa.com
https://github.com/zhudotexe/fanoutqa

6 Ethics Statement

Our question writers and human evaluators were compensated with extra credit in a class they were taking or digital items of their choice, with intrinsic value equivalent to or greater than the time effort spent on our task. Participants gave informed consent and were aware of the compensation before accepting the tasks. Data we collected from human annotators is IRB exempt under 45 CFR 46.104, category 2. No personal identifying information was collected from human participants, and any references to individuals found in the dataset reference publicly-available information (i.e. Wikipedia pages).

Wikipedia text is available under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA) license. We release our dataset under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA) license, and our Python package under the MIT license.

7 Limitations

Due to the limitations of text-based metrics, most of our metrics are biased towards recall over precision. The ROUGE metrics measure precision, but LLMs can output extraneous text that penalizes precision without affecting the factual content of the question. This led to many models scoring high in recall but low in precision, leading to an on-average lower reported F1 score. Although using GPT-4 as a judge model helps measure the factual equivalence of two answers, this may be prohibitively expensive to scale to many more thousands of samples.

FanOutQA uses content solely from English Wikipedia, making it a monolingual dataset. It may be plausible to create parallel datasets using the same provided Wikipedia pages found in other languages, but we leave creation and verification of this dataset to future work.

Acknowledgements

We would like to thank the members of the lab of Chris Callison-Burch for detailed feedback on the contents of this paper and the members of the Northern Lights Province Discord for their participation in our human evaluation. In particular, we would like to thank Bryan Li for his thoughtful suggestions with regards to our human evaluation and other parts of the paper.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Lifeng Shang, Qun Liu, and Kam-Fai Wong. 2023. M4le: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts?

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yuanhua Lv and ChengXiang Zhai. 2011. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 7–16, New York, NY, USA. Association for Computing Machinery.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

OpenAI. 2023. Gpt-4 technical report.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton

Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Andrew Zhu, Liam Dugan, Alyssa Hwang, and Chris Callison-Burch. 2023. Kani: A lightweight and highly hackable framework for building language model applications. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 65–77, Singapore. Association for Computational Linguistics.

A Example Questions

In this section, we provide a sample of various questions found in the FanOutQA dataset, along with their human-written decompositions and answers.

1. **Q:** What is the duration in minutes and seconds of the top 5 songs on the Billboard Year-End Hot 100 singles list of 2022?
   Decomposition:
   (a) **Q:** What are the top 5 songs on the list of Billboard Year-End Hot 100 singles of 2022?
   Evidence:
   `https://en.wikipedia.org/wiki/Billboard_Year-End_Hot_100_singles_of_2022`
   A: Heat Waves, As It Was, Stay, Easy on Me, Shivers
   (b) **Q:** What is the length of Heat Waves?
   Evidence: `https://en.wikipedia.org/wiki/Heat_Waves`
   A: 3:58
   (c) **Q:** What is the length of As It Was?
   Evidence: `https://en.wikipedia.org/wiki/As_It_Was`
   A: 2:43
   (d) **Q:** What is the length of Stay?
   Evidence:
   `https://en.wikipedia.org/wiki/Stay_(The_Kid_Laroi_and_Justin_Bieber_song)`
   A: 2:21
   (e) **Q:** What is the length of Easy on Me?
   Evidence: `https://en.wikipedia.org/wiki/Easy_on_Me`
   A: 3:44
   (f) **Q:** What is the length of Shivers?
   Evidence: `https://en.wikipedia.org/wiki/Shivers_(Ed_Sheeran_song)`
   A: 3:27

   A: `{"Heat Waves": "3:58", "As It Was": "2:43", "Stay": "2:21", "Easy on Me": "3:44", "Shivers": "3:27"}`

2. **Q:** What are the ages of the top 5 most followed people on Instagram?[3]
   Decomposition:
   (a) **Q:** Who are the top 5 most followed on Instagram?
   Evidence:
   `https://en.wikipedia.org/wiki/List_of_most-followed_Instagram_accounts`
   A: Cristiano Ronaldo, Lionel Messi, Selena Gomez, Kylie Jenner, Dwayne Johnson
   (b) **Q:** What is the age of Cristiano Ronaldo?
   Evidence: `https://en.wikipedia.org/wiki/Cristiano_Ronaldo`
   A: 38
   (c) **Q:** What is the age of Lionel Messi?
   Evidence: `https://en.wikipedia.org/wiki/Lionel_Messi`
   A: 36
   (d) **Q:** What is the age of Selena Gomez?
   Evidence: `https://en.wikipedia.org/wiki/Selena_Gomez`
   A: 31
   (e) **Q:** What is the age of Kylie Jenner?
   Evidence: `https://en.wikipedia.org/wiki/Kylie_Jenner`
   A: 26

[3] As of the dataset epoch of Nov 20, 2023. Retrieved documents return the revision as of this date, so answers are consistent over time.

(f) **Q:** What is the age of Dwayne Johnson?
Evidence: https://en.wikipedia.org/wiki/Dwayne_Johnson
A: 51

A: `{ "Cristiano Ronaldo": 38, "Lionel Messi": 36, "Selena Gomez": 31, "Kylie Jenner": 26, "Dwayne Johnson": 51 }`

3. **Q:** What are the top 4 best-selling mangas of all time and who is the protagonist for each?
Decomposition:

(a) **Q:** What are the top 4 best-selling mangas of all time?
Evidence: https://en.wikipedia.org/wiki/List_of_best-selling_manga
A: One Piece, Golgo 13, Case Closed / Detective Conan, Dragon Ball

(b) **Q:** Who is the protagonist of 'One Piece'?
Evidence: https://en.wikipedia.org/wiki/One_Piece
A: Monkey D. Luffy

(c) **Q:** Who is the protagonist of 'Golgo 13'?
Evidence: https://en.wikipedia.org/wiki/Golgo_13
A: Duke Togo

(d) **Q:** Who is the protagonist of 'Case Closed / Detective Conan'?
Evidence: https://en.wikipedia.org/wiki/Case_Closed
A: Shinichi Kudo

(e) **Q:** Who is the protagonist of 'Dragon Ball'?
Evidence: https://en.wikipedia.org/wiki/Dragon_Ball_(manga)
A: Goku

A: `{ "One Piece": "Monkey D. Luffy", "Golgo 13": "Duke Togo", "Case Closed / Detective Conan": "Shinichi Kudo", "Dragon Ball": "Goku" }`

4. **Q:** Among the Ivy League universities, which four have the lowest endowments and how many Nobel laureates do each of them have?
Decomposition:

(a) **Q:** Which 4 Ivy League universities have the lowest endowment?
Evidence: https://en.wikipedia.org/wiki/Ivy_League
A: Brown University, Dartmouth College, Cornell University, Columbia University

(b) **Q:** How many Nobel laureates does Brown University have?
Evidence: https://en.wikipedia.org/wiki/Brown_University
A: 11

(c) **Q:** How many Nobel laureates does Dartmouth College have?
Evidence: https://en.wikipedia.org/wiki/Dartmouth_College
A: 3

(d) **Q:** How many Nobel laureates does Cornell University have?
Evidence: https://en.wikipedia.org/wiki/Cornell_University
A: 62

(e) **Q:** How many Nobel laureates does Columbia University have?
Evidence: https://en.wikipedia.org/wiki/Columbia_University
A: 103

A: `{ "Brown University": 11, "Dartmouth College": 3, "Cornell University": 62, "Columbia University": 103 }`

5. **Q:** What is the area in square kilometers of the city that hosts the alma mater of all partners of the main actors from 'How I Met Your Mother' who eventually hosted the Academy Awards?
Decomposition:

(a) **Q:** Who are the main actors in ‘How I Met Your Mother’?
Evidence: `https://en.wikipedia.org/wiki/How_I_Met_Your_Mother`
A: Josh Radnor, Jason Segel, Cobie Smulders, Neil Patrick Harris, Alyson Hannigan, Cristin Milioti

(b) **Q:** Which of these actors hosted the Academy Awards?
Evidence: `https://en.wikipedia.org/wiki/List_of_Academy_Awards_ceremonies`
A: Neil Patrick Harris

(c) **Q:** Who is the partner of Neil Patrick Harris?
Evidence: `https://en.wikipedia.org/wiki/Neil_Patrick_Harris`
A: David Burtka

(d) **Q:** What is the alma mater of David Burtka?
Evidence: `https://en.wikipedia.org/wiki/David_Burtka`
A: University of Michigan

(e) **Q:** What city is the University of Michigan in?
Evidence: `https://en.wikipedia.org/wiki/University_of_Michigan`
A: Ann Arbor, Michigan

(f) **Q:** What is the area of the city of Ann Arbor?
Evidence: `https://en.wikipedia.org/wiki/Ann_Arbor,_Michigan`
A: 73.35 sq km

A: `73.35 sq km`

6. **Q:** What are the five most popular grape varieties from the Bordeaux appellation, and which area of Bordeaux are they most planted in?
Decomposition:

(a) **Q:** What are the five most popular grape varieties from the Bordeaux appellation?
Evidence: `https://en.wikipedia.org/wiki/Bordeaux_wine`
A: Cabernet Sauvignon, Cabernet Franc, Merlot, Semillon, Sauvignon Blanc

(b) **Q:** Which area of Bordeaux is Cabernet Sauvignon most planted in?
Evidence: `https://en.wikipedia.org/wiki/Cabernet_Sauvignon`
A: Haut-Medoc

(c) **Q:** Which area of Bordeaux is Cabernet Franc most planted in?
Evidence: `https://en.wikipedia.org/wiki/Cabernet_Franc`
A: Saint-Emilion

(d) **Q:** Which area of Bordeaux is Merlot most planted in?
Evidence: `https://en.wikipedia.org/wiki/Merlot`
A: Saint-Emilion and Pomerol

(e) **Q:** Which area of Bordeaux is Semillon most planted in?
Evidence: `https://en.wikipedia.org/wiki/S%C3%A9millon`
A: Saint-Emilion

(f) **Q:** Which area of Bordeaux is Sauvignon Blanc most planted in?
Evidence: `https://en.wikipedia.org/wiki/Sauvignon_blanc`
A: Pessac-Leognan and Graves

A: `{ "Cabernet Sauvignon": "Haut-Medoc", "Cabernet Franc": "Saint-Emilion", "Merlot": "Saint-Emilion and Pomerol", "Semillon": "Saint-Emilion", "Sauvignon Blanc": "Pessac-Leognan and Graves" }`

B Filtering Pipeline

To assess the quality of our dataset and remove unsuitable questions, we used computational methods to identify candidates for removal and manually reviewed them after each round. We started with a heuristic-based algorithm to flag two common indicators of low-quality questions: top-level answers not being composed of sub-question answers and multiple sub-questions using the same Wikipedia article as evidence. Next, we ensured that the knowledge base was being used appropriately by verifying that each sub-question answer is contained in the referenced article. Since Wikipedia is a large resource and the writers may not have seen every article related to their questions, we used the OpenAI embeddings (`text-embedding-3-small`, henceforth "embeddings"; Neelakantan et al., 2022) of top-level questions and article titles to retrieve the 30 most similar Wikipedia articles for each question. If any of these articles contained all answers to the sub-questions, we removed the entire example from the dataset. This ensures that the questions both can and need to be answered by the fan-out method.

In the final round of reviewing the quality of our dataset, we used GPT-4 (`gpt-4-0613`) with greedy sampling to help remove or fix poorly phrased questions (prompts in Appendix F). We prompted GPT-4 to identify if a question is not objective, such as "What are five inventions in the Industrial Revolution?" or "Who are the five most famous celebrities?" It was also instructed to identify questions that were missing numeric units and suggest grammar corrections. We manually reviewed all LLM-assisted modifications before deduplication. Finally, we considered duplicate questions to have embeddings with cosine similarity within 0.9. We manually reviewed these duplicates and selected one to remain in the final dataset.

C Models Used

We benchmarked the following state-of-the-art LLMs' performance on FanOutQA. Where needed, the specific model's key/sub-version is provided.

Commercial Models

- GPT-4 (`gpt-4-0613`, OpenAI, 2023)
- GPT-4-turbo (`gpt-4-0125-preview`[4])
- GPT-3.5-turbo (`gpt-3.5-turbo-1106`[5])
- Claude (`claude-2.1`[6])

Open-Source Models

- LLaMA 2 (`Llama-2-70b-chat`, Touvron et al., 2023)
- Mistral 7B (`Mistral-7B-Instruct-v0.2`, Jiang et al., 2023)
- Mixtral 8x7B (`Mixtral-8x7B-Instruct-v0.1`, Jiang et al., 2024)

All models were sampled using greedy decoding, and local models were loaded using FP16 precision on 3 NVIDIA RTX A6000s. We provided the seed `31415` to OpenAI's GPT models for deterministic generation.

D Results Table

We tabulate the results of each model and metric in Table 1. We also run additional experiments in which we fix the context size of each model to be equal to the shortest model's (4096 tokens) to verify correlations between context length and performance, the results of which we tabulate in Table 2.

[4] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

[5] https://platform.openai.com/docs/models/gpt-3-5-turbo

[6] https://www.anthropic.com/news/claude-2-1

Closed Book

Model	Ctx Size	Loose	Strict	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	GPT Judge
LLaMA 2	4,096	0.440	0.058	0.285	0.149	0.238	0.441	0.120
GPT-4	8,096	0.355	0.066	0.313	0.177	0.267	0.419	0.149
GPT-3.5-turbo	16,384	0.398	0.058	0.401	0.227	0.342	0.455	0.145
Mistral-7B	32,768	0.427	0.055	0.260	0.123	0.212	0.449	0.102
Mixtral-8x7B	32,768	**0.470**	0.081	0.302	0.158	0.254	0.466	0.186
GPT-4-turbo	128,000	0.460	**0.101**	**0.482**	**0.290**	**0.409**	**0.493**	**0.199**
Claude 2.1	200,000	0.341	0.041	0.412	0.208	0.344	0.426	0.110

Open Book

Model	Ctx Size	Loose	Strict	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	GPT Judge
LLaMA 2	4,096	0.390	0.064	0.157	0.075	0.131	0.443	0.108
GPT-4	8,096	0.315	0.057	0.208	0.106	0.183	0.427	0.164
GPT-3.5-turbo	16,384	0.155	0.032	0.114	0.051	0.099	0.338	0.076
Mistral-7B	32,768	—	—	—	—	—	—	—
Mixtral-8x7B	32,768	0.396	0.055	0.173	0.078	0.147	0.449	0.148
GPT-4-turbo	128,000	0.470	**0.109**	**0.356**	**0.207**	**0.314**	**0.487**	**0.262**
Claude 2.1	200,000	**0.471**	0.086	0.295	0.157	0.253	0.485	0.218
Human	—	0.685	0.289	0.344	0.210	0.307	0.413	0.452

Evidence Provided

Model	Ctx Size	Loose	Strict	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	GPT Judge
LLaMA 2	4,096	0.514	0.077	0.376	0.206	0.304	0.472	0.162
GPT-4	8,096	0.546	0.144	0.500	0.301	0.413	0.530	0.304
GPT-3.5-turbo	16,384	0.517	0.102	0.455	0.252	0.358	0.497	0.243
Mistral-7B	32,768	0.540	0.088	0.330	0.172	0.264	0.475	0.202
Mixtral-8x7B	32,768	0.576	0.135	0.409	0.231	0.343	0.509	0.283
GPT-4-turbo	128,000	0.628	0.192	**0.614**	**0.395**	**0.523**	**0.581**	0.413
Claude 2.1	200,000	**0.653**	**0.215**	0.423	0.262	0.354	0.508	**0.470**

Table 1: Performance of each model on all metrics and all settings. We include human performance in the open-book setting, and omit Mistral-7B's performance in the open-book setting due to catastrophic neural text degeneration.

Open Book, Context Limited

Model	Ctx Size	Loose	Strict	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	GPT Judge
LLaMA 2	4,096	0.423	0.066	0.194	0.095	0.163	0.449	0.113
GPT-4	4,096	0.236	0.040	0.151	0.071	0.134	0.395	0.102
GPT-3.5-turbo	4,096	0.124	0.023	0.099	0.041	0.087	0.326	0.054
Mistral-7B	4,096	—	—	—	—	—	—	—
Mixtral-8x7B	4,096	**0.458**	**0.076**	**0.224**	0.105	**0.192**	**0.465**	**0.160**
GPT-4-turbo	4,096	0.294	0.051	0.194	0.103	0.169	0.427	0.137
Claude 2.1	4,096	0.348	0.055	**0.224**	**0.113**	0.187	0.445	0.140

Evidence Provided, Context Limited

Model	Ctx Size	Loose	Strict	ROUGE-1	ROUGE-2	ROUGE-L	BLEURT	GPT Judge
LLaMA 2	4,096	0.514	0.077	**0.376**	0.206	0.304	0.472	0.160
GPT-4	4,096	0.380	0.083	0.157	0.075	0.131	0.443	0.184
GPT-3.5-turbo	4,096	0.425	0.054	0.208	0.106	0.183	0.427	0.162
Mistral-7B	4,096	0.466	0.040	0.114	0.051	0.099	0.338	0.134
Mixtral-8x7B	4,096	**0.525**	0.102	0.173	0.078	0.147	0.449	0.229
GPT-4-turbo	4,096	0.515	**0.113**	0.356	**0.207**	**0.314**	**0.487**	**0.250**
Claude 2.1	4,096	0.490	0.084	0.295	0.157	0.253	0.485	0.189

Table 2: Performance of each model with a fixed context length on all metrics in the open-book and evidence-provided settings. We omit Mistral-7B's performance in the open-book setting due to catastrophic neural text degeneration.

E Human Instructions

E.1 Question Writing Instructions

We presented the following instructions to students in a Google Colaboratory notebook. To write the questions and their decompositions, students wrote them as a Python dictionary, which the notebook validated the structure of before their submission. The remainder of this section contains the verbatim instructions included in the notebook.

We are creating a challenge problem for natural language processing systems, where systems have to answer questions that require them to read multiple sources.

Specifically, we're looking at "fan-out" questions - where the question itself is not too long, but to answer it requires first looking up (or being supplied) some list of items, then finding out more details about each item.

Your job is to help us write:

- these fan-out questions
- strategies to answer the questions you write, with relevant Wikipedia articles linked
- reference answers to these questions.

You'll be using this Colab notebook to make sure the questions and answers are in the right format. Let's take a look at a couple examples, first:

For example, a very simple fan-out question might be:

> What was the population of New York and Los Angeles in 1950?

In this example, the best strategy to answer this question is to split it once into two questions, "What was the population of New York in 1950?" and "What was the population of Los Angeles in 1950"?

```
# EXAMPLE FORMAT - DO NOT MODIFY
example_q1 = {
  "question": "What was the population of New York and Los Angeles in 1950?",
  "strategy": [
    # each question in here is the same structure recursively!
    # we don't need to here, but subquestions can be broken up even further
    {
      "question": "What was the population of New York in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/
    Demographic_history_of_New_York_City",
      "answer": 7891957
    },
    {
      "question": "What was the population of Los Angeles in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/Los_Angeles",
      "answer": 1970358
    },
  ],
  "answer": {
    "New York": 7891957,
    "Los Angeles": 1970358
  }
}

validate_question(example_q1, is_demonstration=True)
# END EXAMPLE 1
```

We can make this question more complex by making the system look up the list of items rather than providing it in the question:

> What was the population in 1950 of the 5 current most populous cities in the United States?

Now, to answer the question, one has to first look up a list of populous cities in the US (the *strategy*), then fan-out based on that information.

```
# EXAMPLE FORMAT - DO NOT MODIFY
example_q2 = {
  "question": "What was the population in 1950 of the 5 current most populous cities
     in the United States?",
  # use "strategy" for questions that don't depend on the answers to previous
    questions
  "strategy": [
    {
      "question": "What are the 5 most populous cities in the United States?",
      "evidence": "https://en.wikipedia.org/wiki/
    List_of_United_States_cities_by_population",
      "answer": ["New York", "Los Angeles", "Chicago", "Houston", "Phoenix"]
    },
  ],
  # use "then" if sub-questions depend on answers to the questions in "strategy"
  "then": [
    {
      "question": "What was the population of New York in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/
    Demographic_history_of_New_York_City",
      "answer": 7891957
    },
    {
      "question": "What was the population of Los Angeles in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/Los_Angeles",
      "answer": 1970358
    },
    {
      "question": "What was the population of Chicago in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/Chicago",
      "answer": 3620962
    },
    {
      "question": "What was the population of Houston in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/Houston",
      "answer": 596163
    },
    {
      "question": "What was the population of Phoenix in 1950?",
      "evidence": "https://en.wikipedia.org/wiki/Phoenix,_Arizona",
      "answer": 106818
    },
  ],
  "answer": {
    "New York": 7891957,
    "Los Angeles": 1970358,
    "Chicago": 3620962,
    "Houston": 596163,
    "Phoenix": 106818
  }
}

validate_question(example_q2)
# END EXAMPLE 2
```

Let's look at one more example that's a bit more complex. We'll ask the question:

> Find the female cabinet members of the current US President. Who are those cabinet members and what city/town were they born in?

Now, we need to look up quite a bit more information:

```
# EXAMPLE FORMAT - DO NOT MODIFY
example_q3 = {
  "question": "Find the female cabinet members of the current US President. Who are
    those cabinet members and what city/town were they born in?",
  "strategy": [
    {
      "question": "Who is the current US President?",
      "evidence": "https://en.wikipedia.org/wiki/
    List_of_presidents_of_the_United_States",
      "answer": "Joe Biden",
    }
  ],
  "then": [
    {
      "question": "Who are the female members of Joe Biden's cabinet and what city/
    town were they born in?",
      "strategy": [
        {
          "question": "Who are the female members of Joe Biden's cabinet?",
          "evidence": "https://en.wikipedia.org/wiki/Cabinet_of_Joe_Biden",
          "answer": ["Kamala Harris", "Janet Yellen", "Deb Haaland", "Gina Raimondo"
    , "Julie Su", "Marcia Fudge", "Jennifer Granholm"]
        }
      ],
      "then": [
        {
          "question": "What city/town was Kamala Harris born in?",
          "evidence": "https://en.wikipedia.org/wiki/Kamala_Harris",
          "answer": "Oakland, California"
        },
        {
          "question": "What city/town was Janet Yellen born in?",
          "evidence": "https://en.wikipedia.org/wiki/Janet_Yellen",
          "answer": "New York City, New York"
        },
        {
          "question": "What city/town was Deb Haaland born in?",
          "evidence": "https://en.wikipedia.org/wiki/Deb_Haaland",
          "answer": "Winslow, Arizona"
        },
        {
          "question": "What city/town was Gina Raimondo born in?",
          "evidence": "https://en.wikipedia.org/wiki/Gina_Raimondo",
          "answer": "Smithfield, Rhode Island"
        },
        {
          "question": "What city/town was Julie Su born in?",
          "evidence": "https://en.wikipedia.org/wiki/Julie_Su",
          "answer": "Madison, Wisconsin"
        },
        {
          "question": "What city/town was Marcia Fudge born in?",
          "evidence": "https://en.wikipedia.org/wiki/Marcia_Fudge",
          "answer": "Cleveland, Ohio"
        },
        {
          "question": "What city/town was Jennifer Granholm born in?",
          "evidence": "https://en.wikipedia.org/wiki/Jennifer_Granholm",
          "answer": "Vancouver, British Colombia"
        },
      ],
      "answer": {
        "Kamala Harris": "Oakland, California",
        "Janet Yellen": "New York City, New York",
        "Deb Haaland": "Winslow, Arizona",
        "Gina Raimondo": "Smithfield, Rhode Island",
        "Julie Su": "Madison, Wisconsin",
        "Marcia Fudge": "Cleveland, Ohio",
        "Jennifer Granholm": "Vancouver, British Colombia"
      }
    }
  ],
  "answer": {
    "Kamala Harris": "Oakland, California",
    "Janet Yellen": "New York City, New York",
    "Deb Haaland": "Winslow, Arizona",
    "Gina Raimondo": "Smithfield, Rhode Island",
    "Julie Su": "Madison, Wisconsin",
    "Marcia Fudge": "Cleveland, Ohio",
    "Jennifer Granholm": "Vancouver, British Colombia"
  },
}

validate_question(example_q3)
# END EXAMPLE 3
```

Now it's up to you to write 1-5 of these questions in the format provided!

The questions can be about any topic where information is available on English Wikipedia - it does not necessarily have to be related to the class. Your evidence should be a link to a single page on English Wikipedia. Try to make your questions fairly diverse and unambiguous (e.g. include the units the answer is expected in, if applicable).

The answer to a top-level question must not be available on a singular Wikipedia article. Your question must require looking at at least 5 Wikipedia articles.

If your question does not validate, please read the error to see what changes are needed.

Use this template for each question/subquestion:

```
{
  "question": "YOUR QUESTION HERE",
  "strategy": [
    # subquestions
  ],
  "then": [
    # more subquestions that depend on answering the questions in "strategy" first (
    if any)
  ],
  "evidence": "link to wikipedia",  # each subquestion needs evidence to answer it,
    or a recursive strategy - you should either have evidence or strategy, but not
    both
  "answer": 0  # can be a dict, list, or primitive value
}
```

Glossary

`question` (str): The question to be answered. At the root node, this should not be answerable without breaking it up into smaller subquestions.

`strategy` (list of Question): Subquestions to break the question up into. These shouldn't require looking anything up to ask (e.g. see example 1 vs 2).

`then` (list of Question, optional): Subquestions to ask with the information gathered after answering all the subquestions in strategy, if any are needed.

`evidence` (link to Wikipedia): If question can be answered by information found on a single Wikipedia page, the link to that page.

`answer` (dict, list, or primitive): The final answer to the question, after all subquestions have been answered.

Tip: Either evidence or strategy may be present in a subquestion, but not both. If the answer to a question can be found on a single Wikipedia page, use evidence. If you need to break it up into smaller questions, use strategy (and possibly then).

There might be multiple valid strategies to answer a top-level question; use the one that is most intuitive to you. After writing your question, validate it with `validate_question` and see if it makes sense to read.

Blank code cells follow for question writing.

E.2 Question Answering Instructions

We presented the following instructions to volunteers participating in our human evaluation after they gave their informed consent. These instructions imitate the Open Book setting for models.

Thanks for participating in the FanOutQA human evaluation! You will be given 10 questions, and your task is to answer the questions to the best of your ability.

You may use English Wikipedia (https://en.wikipedia.org/wiki/Main_Page) to search for Wikipedia articles to help you answer each question. **Do not use Google or other search engines.** Please record which Wikipedia articles you looked at (whether or not you used the information in the article) to answer the questions.

To answer the questions, please make a copy of this Google doc, and fill in your answers in the spaces below. Once you are finished, please send the document as a PDF to <first author's email>.

- Answers do not need to be complete sentences.
- Answers do not need to be in a particular format - they will be judged by a human.
- Some questions may only require a single answer, others may need a list.
- You do not need to finish all 10 questions in a single sitting.
- You will be awarded based on the number of questions completed, regardless of whether or not the answer is correct. Please do your best to answer correctly though! You will not be given an award if the answers are obviously low-effort.

A list of ten questions, randomly sampled from the FanOutQA test set per participant, follows.

F LLM Prompts

F.1 Subjective Flag

```
SYSTEM: You are assessing how well a given question can be answered. For each
    submission, assess whether the provided question can be answered
    deterministically and objectively at a fixed point in time as of January 2024
    given access to appropriate information sources.

USER: [Question]: {question}
***
Can the question be answered in a way that is both deterministic (i.e., the answer
    has a single unambiguously correct answer) and objective (i.e., the answer is
    based on factual information and not influenced by personal feelings or opinions
    ) at a given point in time? If the question allows for multiple correct answers,
     it should not be considered deterministic.
For each question, provide a step-by-step reasoning for your assessment before your
    conclusion, then print only the single character "Y" or "N" (without quotes or
    punctuation) on its own line corresponding to the correct answer. At the end,
    repeat just the letter again by itself on a new line.
```

If the model's response ended with the letter "N", we flagged the question for manual review.

F.2 Grammaticality and Unit Suggestions

```
SYSTEM: You are assessing how well a given question can be answered. For each
    question and answer, assess whether the question is grammatical and includes the
     expected units (if applicable).
If the question does not require any changes, output "No change."
Otherwise, rewrite the question to make it grammatical and include any necessary
    units without changing the provided answer. Output only the rewrite.
If this is not possible, output the word "FLAG" on its own line, followed by your
    reasoning.

USER: [Question]: {question}
***
[Answer]: {answer}
```

If the model's response began with "FLAG", we recorded the response for manual review. Otherwise, if the model's response was not "No change.", we recorded the suggested rewrite. Afterwards, we manually reviewed all suggestions made by the model.

F.3 Model Judge

```
SYSTEM: You are comparing a submitted answer to an expert answer on a given question

USER: [BEGIN DATA]
************
[Question]: {question}
************
[Expert]: {reference}
************
[Submission]: {answer}
************
[END DATA]

Compare the factual content of the submitted answer with the expert answer. Ignore
    any differences in style, grammar, or punctuation.
The submitted answer may either be a subset or superset of the expert answer, or it
    may conflict with it. Determine which case applies. First, write out in a step
    by step manner your reasoning about the factual content to be sure that your
    conclusion is correct. Avoid simply stating the correct answers at the outset.
    Then print only the single character "A", "B", "C", "D", "E", or "F" (without
    quotes or punctuation) on its own line corresponding to the correct answer. At
    the end, repeat just the letter again by itself on a new line.
(A) The submitted answer is a subset of the expert answer and is fully consistent
    with it.
(B) The submitted answer is a superset of the expert answer and is fully consistent
    with it.
(C) The submitted answer contains all the same details as the expert answer.
(D) There is a disagreement between the submitted answer and the expert answer.
(E) The answers differ, but these differences don't matter from the perspective of
    factuality.
(F) The submitted answer does not answer the question or is otherwise invalid.
```

If the model's response ended with the letter "B", "C", or "E", we awarded the answer a score of 1.0. Otherwise, we awarded the answer a score of 0.0.