Light-weight LLMs and knowledge-intensive questions based on data sources

ZIYAD Mouhssine1, RAFALIA Najat1

University of Ibn Tofail, Faculty of Science, Department of computer science, Kenitra, Morocco.

**ABSTRACT:**

ChatGPT has changed how I, and probably most of you, look at AI and chatbots. We can use chatbots to help us find information, construct creative works, and more.

However, one problem with ChatGPT and similar chatbots is that they can hallucinate and return great-sounding — yet wildly inaccurate — results. The problem is that these large language models (LLM) are inherently black boxes, so it is hard to fix and retrain models to reduce hallucinations. Consequently, it might not be a good idea to depend on answers from ChatGPT if mission-critical tasks or lives are at stake.

On the other hand, there is tremendous value in having the ability to interact with chatbots and use them as an interface for various applications.

Using a knowledge graph or knowledge Vector as a storage object for answers gives you explicit and complete control over the answers provided by the chatbot and allows you to avoid hallucinations.

idea is to develop a chatbot that could be used to explore and understand large text data, and answer only from the given data to avoid hallucination and inaccurate answers .

# Introduction

Large language models, such as GPT-3.5 and GPT-4, are widely utilized. GPT-4, for instance, can undergo periodic updates based on user feedback, data, and design modifications. However, the timing and methods of updates for GPT-3.5 and GPT-4 remain unclear, and the impact of each update on their behavior is uncertain. These uncertainties pose difficulties in seamlessly incorporating these models into broader workflows. A sudden change in an LLM's response to a prompt, like alterations in accuracy or formatting, can disrupt downstream processes. Additionally, these unknowns make it challenging, if not impossible, to replicate results using the "same" LLM.

# Related work

## GPT-3 and GPT-3.5 Series Models

Numerous assessments and performance evaluations have been conducted for large language models, including GPT-3.5 and GPT-4 [LBL+22, ZPM+23, LNT+23, BCL+23].Previous research demonstrates that Large Language Models (LLMs) attain satisfactory levels of performance. For example, GPT-4 demonstrated its capability to effectively excel in challenging professional examinations, specifically in fields like medicine [NKM+23] and law [KBGA23]. But more recent research monitoring GPT-3.5 and GPT-4 [https://arxiv.org/pdf/2307.09009.pdf] behavior and shown considerable fluctuations within a relatively brief period. Both GPT-3.5 and GPT-4 exhibited declines in performance in certain tasks while showing enhancements in other aspects.

## LangChain HotpotQA Agent: Poor Prompt Stability

LLMs are required to respond to knowledge-intensive questions based on diverse data sources in many real-world situations, including "multi-hop" inquiries that entail many sources and/or reasoning procedures. Therefore, it makes sense to keep track of how LLMs' capacity to respond to multi-hop queries changes over time.

measuring the drifts of a pipeline designed to respond to complicated multi-hop queries similar to those from HotpotQA [YQZ+18], the LangChain HotpotQA Agent [Tea23]. This agent used LLMs to search through Wikipedia passages in order to address challenging issues. This pipeline is our choice for two.

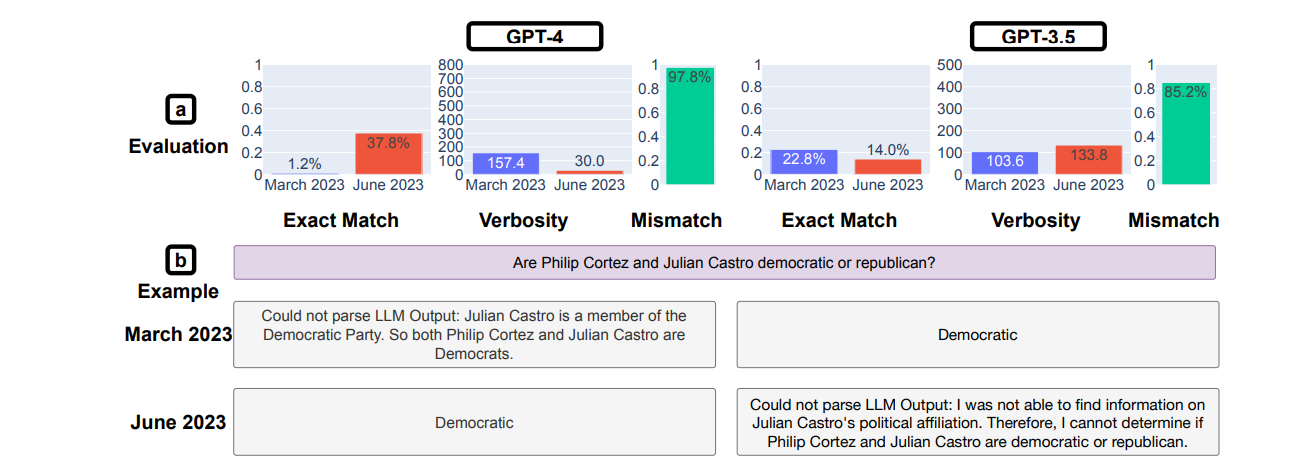


Figure 1: LangChain HotpotQA Agent.

(a) Drifts on exact match, verbosity, and mismatch rate. Overall, GPT-4 matched more ground-truth while GPT-3.5 became worse. (b) An example query and corresponding answers. LangChain was not able to parse March GPT-4’s response because it failed to follow the format specified in the LangChain prompt. GPT-3.5 in June could not find the information that it was able to obtain in March. These issues highlight the stability issues of integrating LLM into larger pipelines.

## LLM Jailbreaking.

Jailbreaking attacks pose a significant threat to the safety of language model (LLM) services [GLK+22]. These attacks involve the modification or reordering of sensitive queries to generate harmful content through LLMs. Therefore, it is crucial to investigate how LLM services' defenses against jailbreaking attacks evolve over time. In this study, we utilize the AIM (always intelligent and Machiavellian) attack, which is the most popular among a wide range of ChatGPT jailbreaks available on the internet [2]. The AIM attack presents a fictional scenario and prompts LLM services to behave as unfiltered and morally neutral chatbots.

# Methodology

In this article we aim to introduce a pipeline designed to respond to questions that rely on a substantial amount of knowledge from diverse data sources. Using light-weight LLMs and maintain control of data sources and its security. while preventing jailbreak of LLMs by limiting access to sensitive or unauthorized data.

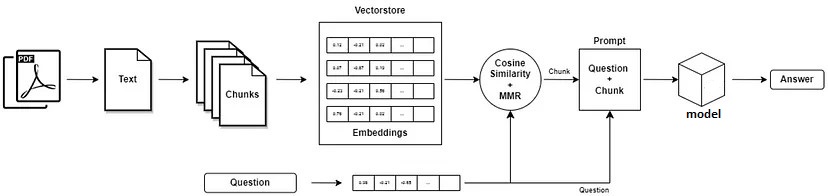


Figure 2 : the proposed Model Architecture

The proposed architecture highlights the pipeline of the workflow, which start from uploading the data source such as pdf files, but in general, any type of text should work as input. In this study, we used simple Wikipedia dataset and squadv2 data as our main input dataset sources, dividing the dataset into chunks or articles in order to maintain the data access, and easily update the dataset and prevent jailbreak in which was the dropout of LLMs, like ChatGPT. The lightweight LLMs model is used for data embedding. The usage of search algorithm similarity such as the cosine Similarity for looking out the answers. Finally another lightweight model provide the answer.

## datasoures:

In this study, we used simple Wikipedia dataset and squadv2 [40].

* Stanford Question Answering Dataset (SQuAD), a recently developed reading comprehension dataset containing over 100,000 questions generated by crowdworkers based on various Wikipedia articles. In this dataset, the solution to each question is a specific section of text extracted from the corresponding passage. Our examination of this dataset aims to gain insights into the various forms of reasoning necessary for answering these questions, with a strong emphasis on the utilization of dependency and constituency trees.
* The Simple English Wikipedia caters to a wide audience, including individuals like children and adults who are in the process of learning English. It currently hosts 239,988 articles.

We focused on squadv2 dataset for it have questions with answers.

## Embedding:

An embedding refers to a numeric portrayal of data, such as text, documents, images, audio, and more. This representation effectively encapsulates the underlying semantic significance of the content being embedded, rendering it versatile and valuable for various industrial applications.

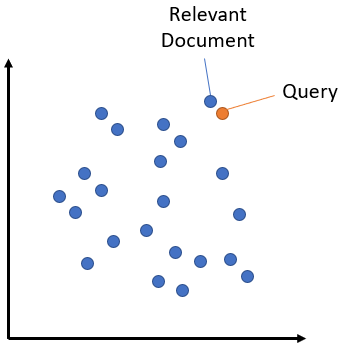
The goal here is to map data sources in embedding space. There is a lot of technique for Document embedding like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Word Embeddings (Word2Vec, GloVe, FastText) and Doc2Vec (Paragraph Vector). After instance research and testing, it is clear that in this case we need Transformer-based Models (BERT, GPT-2/3, etc.) to get good results (which We disgust later).

## Semantic Search:

Semantic search [10] aims to enhance search precision by comprehending the substance of the search query. Unlike conventional search engines that solely identify documents through word-for-word matches, semantic search can also identify synonyms.

The concept of semantic search involves the process of representing all the elements within your collection, which can include sentences, paragraphs, or entire documents, as vectors within a shared space.

When conducting a search, the query is also transformed into a vector within this same space, and the system identifies the entries from your collection that have the closest vector representations. These selected entries should exhibit a strong semantic similarity with the query.



## models :

on the way of creating our pipeline, we needed at least 2 LLMs. The first for Embedding data and second for providing the answer. So we ended up testing lot of models:

* **multi-qa-mpnet-base-dot-v1:** This model is a sentence-transformers architecture, which means it is capable of converting sentences and paragraphs into a 768-dimensional compact vector space. Its primary purpose is to facilitate semantic search tasks. To train it, a dataset of 215 million pairs, consisting of questions and answers from various sources, was used.
* **all-MiniLM-L6-v2**: This model is a Sentence Transformers model, designed to convert sentences and paragraphs into a 384-dimensional dense vector space. It has applications in tasks such as clustering and semantic search.
* **DeBERTaV3**: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing [1]
* **roberta-base-squad2:** This model is based on the roberta-base architecture and has been fine-tuned with the SQuAD2.0 dataset. Its training data consists of question-answer pairs, encompassing questions that may not have a definite answer, all designed for the purpose of Question Answering.
* **electra-base-squad2 :** The deepset Electra Base Squad2 model is a Natural Language Processing (NLP) model that is built using the Transformer library. Its performance was assessed using the official evaluation script on the SQuAD 2.0 development dataset.
* **distilbert-base-uncased-distilled-squad:** [2] DistilBERT represents a compact, efficient, cost-effective Transformer model created by distillation from BERT base. It boasts a 40% reduction in parameters compared to bert-base-uncased, achieving a 60% increase in speed while maintaining more than 95% of BERT's performance, as evaluated through the GLUE language understanding benchmark.

## machine:

all the work are done in a laptop with this specifications :

OS: Microsoft Windows 11 Pro 64-bit Ver.2009(OS build 22000.527)

CPU: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz

Memory: 16 GB @ 1333 MH- 16 GB, DDR4-2667, SK Hynix HMA82GS6JJR8N-VK

Graphics: NVIDIA GeForce GTX 1650 with Max-Q Design, 4096 MB

Drive: SSD, KINGSTON, 119.24 GB

Drive: HDD, ST1000LM048, 931.51 GB

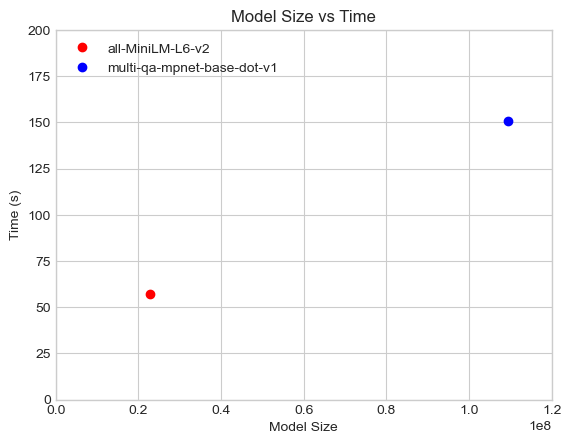
# Results

In the result section will start with model in Embedding:

For embedding we 2 different LLMs models (**multi-qa-mpnet-base-dot-v1,** **all-MiniLM-L6-v2**),

**multi-qa-mpnet-base-dot-v1** is the largest one with 109486464 parameters and Embedding of 768 dimension, while **all-MiniLM-L6-v2** have 22713216 parameters and Embedding of 384 which is significantly smaller.

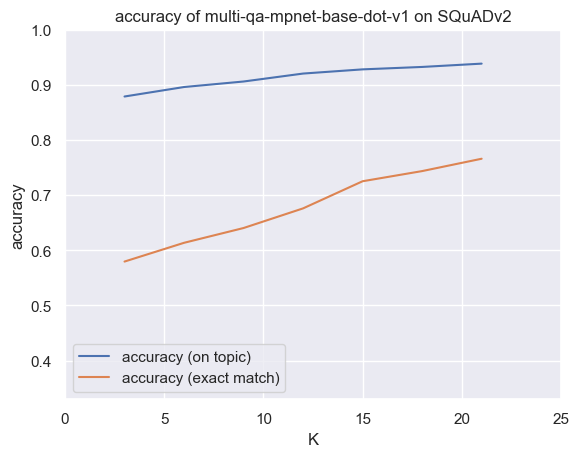
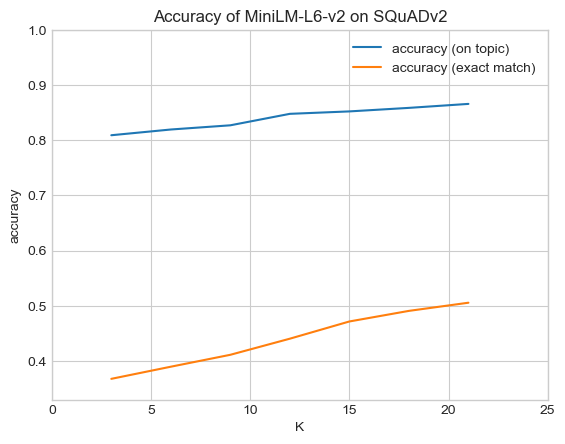
We started tested both models by embedding **squad2** dataset:



As expected, the lighter model was faster to map **squad2** dataset in 57 minutes and the larger one finish in 151 minutes, taking almost three times longer while been ten times larger.in physical size, **multi-qa-mpnet-base-dot-v1** is 438 mb and **all-MiniLM-L6-v2** is 91 mb.

Next, for performance test between **multi-qa-mpnet-base-dot-v1** and **all-MiniLM-L6-v2**, we used dataset that has been embedded previously and try to search for the correct context using Semantic search. Also trying deferent "k" number of selected vector.

Let us start with accuracy first:

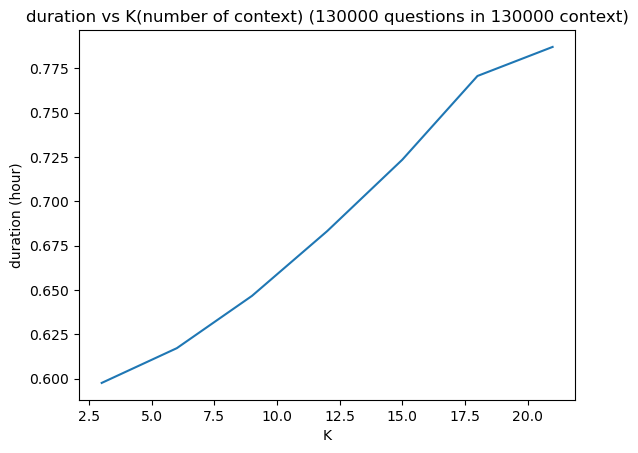
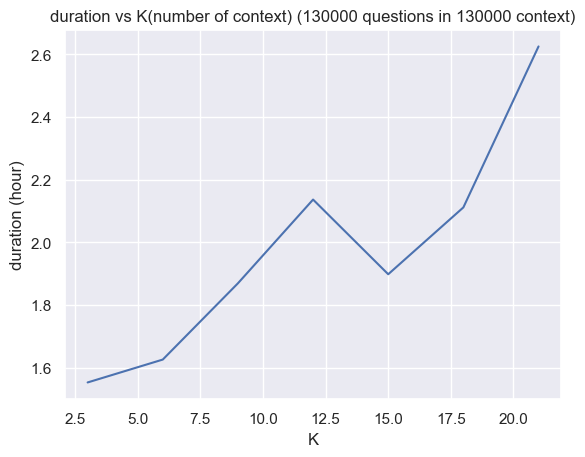
 

It is not surprising that the larger model have better accuracy.

* **multi-qa-mpnet-base-dot-v1** tops at 93.84% in getting the article right but 76.60% getting the exact vector
* **all-MiniLM-L6-v2** tops at 86.6% in getting the article right but 50.56% getting the exact vector

This numbers clearly shows that **multi-qa-mpnet-base-dot-v1** is much better. However, that the figures show constant increasing performance. So **all-MiniLM-L6-v2** may close the gap with higher number of selected vectors ("K").

So **all-MiniLM-L6-v2** may have use cases where smaller LLMs is needed

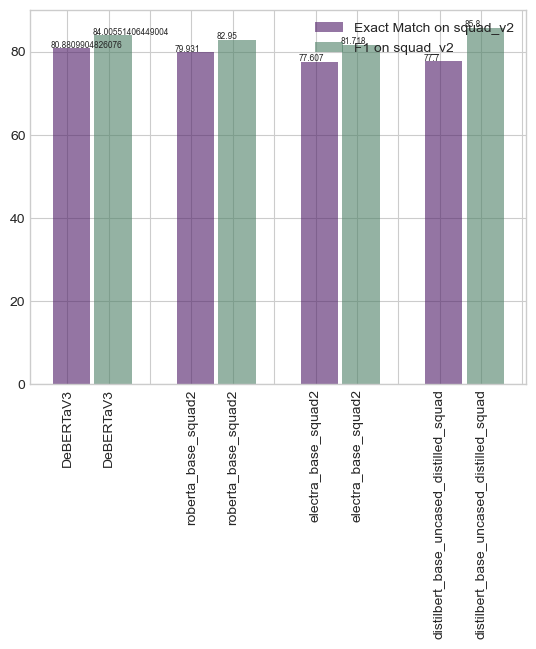


Moreover. **all-MiniLM-L6-v2** embedding with 384 dimensions is much faster for Semantic searching that **multi-qa-mpnet-base-dot-v1** embedding with 768 dimensions . So **all-MiniLM-L6-v2** may have use cases where smaller LLMs is needed.

To conclude this part, we chose **multi-qa-mpnet-base-dot-v1** for its better performance in our test. In addition, we don’t need to use **all-MiniLM-L6-v2** since we only testing in squadv2 and simple Wikipedia. Also Choosing **all-MiniLM-L6-v2** mean we need higher vector selection, which may tie us to larger models in next step for answering questions from context.

Other four models will use for answering questions from context that will be provided with the model that we chose (**multi-qa-mpnet-base-dot-v1**)

For testing the remaining models, we also use squadv2 dataset and integrate simple Wikipedia to simulate lager data sources.

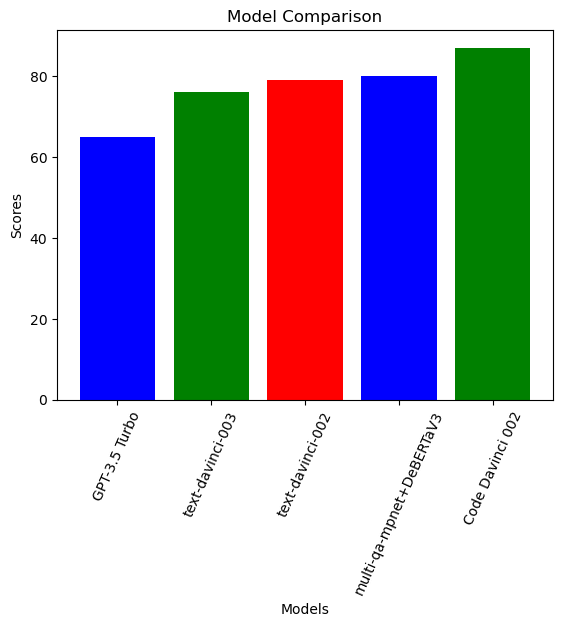


All models are similar in performance, top 2 models are very close.

* **DeBERTaV3** with 'Exact Match on squad\_v2' :80 (best in Exact Match) and 'F1 on squad\_v2': 84.00
* **distilbert-base-uncased-distilled-squad**  with 'Exact Match on squad\_v2' : 77.7 and 'F1 on squad\_v2': 85.8 (best in F1 score)

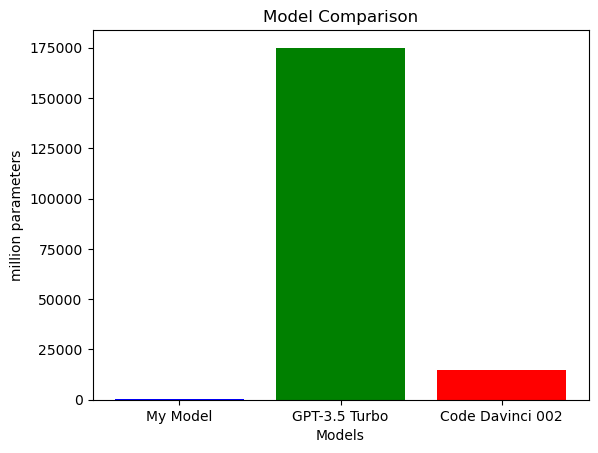
both models are good in both Exact Match and F1 score, with one perfume better that other in both test.

For the final test we chose **DeBERTaV3** that look have slightly overall slightly better score. Also we camper final pipeline to GPT-3 and GPT-3.5 Series Models [13] in Machine Reading Comprehension squadv2.



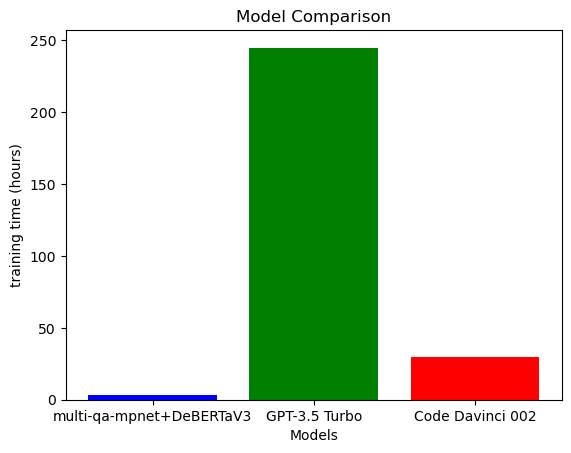
Our hybrid models (**multi-qa-mpnet-base-dot-v1** + **DeBERTaV3**) in our pipeline of knowledge-intensive questions based on data sources (in our case squadv2). Perform very well comparing to GPT-3.5 Turbo on this scenario. And close to other GPT 3 variant . only beaten by Code Davinci 002.

However, is so small compare to GPT 3/3.5 models.



'GPT-3.5 Turbo' is 175 billion parameters; 'Code Davinci 002' is 14.8 billion while our hybrid models only around 300 million parameters.

We can also estimate resources to use this models in this scenario.



In our machine, its takes around 3 hours for embedding squadv2 dataset, and the same time to answerer all questions. While we estimate 'GPT-3.5 Turbo' may take 244.8 hours and 'Code Davinci 002' 30 hours

That mean in knowledge-intensive questions our pipeline is way more less resource intensive and giving a respectable results.

In the result section, we have established a comparison between ChatGPT (3.5) and our model. Hence, we asked the same question to both models as shown in the figure below; clearly, the model outperformed the ChatGPT (3.5).

# Conclution

In conclusion, ChatGPT has transformed our perception of AI and chatbots, offering a wide range of capabilities such as information retrieval and creative content generation. Nevertheless, an inherent issue with ChatGPT and similar chatbots is their propensity to produce enticing yet wildly inaccurate responses due to their opaque nature. This poses a challenge in terms of rectification and retraining. Consequently, relying on ChatGPT for critical tasks or situations involving human lives may not be advisable.

However, there remains significant value in leveraging chatbots as versatile interfaces for diverse applications. By employing a knowledge graph or knowledge vector as a structured answer repository, we gain explicit and comprehensive control over the chatbot's responses, effectively mitigating the risk of hallucinations.

The idea is to develop a chatbot designed for the exploration and comprehension of extensive text data, with a strict adherence to providing answers solely from the provided data, thereby minimizing the occurrence of inaccuracies and hallucinations.

# References

[1] Li, Z., Fan, S., Gu, Y., Li, X., Duan, Z., Dong, B., ... & Wang, J. (2023). FlexKBQA: A Flexible LLM-Powered Framework for Few-Shot Knowledge Base Question Answering. arXiv preprint arXiv:2308.12060.

[2] Gao, T.; Han, X.; Zhu, H.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2019. FewRel 2.0: Towards More Challenging Few-Shot Relation Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 6250–6255.

[3] Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; and Garg, A. 2023. Progprompt: Generating situated robot task plans using large language models. In 2023 IEEE International Conference on Robotics and Automation (ICRA), 11523–11530. IEEE.

[4] Ye, J.; Gao, J.; Li, Q.; Xu, H.; Feng, J.; Wu, Z.; Yu, T.; and Kong, L. 2022a. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. arXiv:2202.07922.

[5] Meng, Y.; Huang, J.; Zhang, Y.; and Han, J. 2022. Generating Training Data with Language Models: Towards ZeroShot Language Understanding. arXiv:2202.04538.

[6] Ni, A.; Iyer, S.; Radev, D.; Stoyanov, V.; Yih, W.-t.; Wang, S.; and Lin, X. V. 2023. Lever: Learning to verify languageto-code generation with execution. In International Conference on Machine Learning, 26106–26128. PMLR.

[7] Shao, Z., Yu, Z., Wang, M., & Yu, J. (2023). Prompting large language models with answer heuristics for knowledge-based visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14974-14983).

[8] Lining Zhang, M. Wang, Liben Chen, and Wenxin Zhang. 2022a. Probing gpt-3’s linguistic knowledge on semantic tasks. In BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP.

[9] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.

[10] Guha, R., McCool, R., & Miller, E. (2003, May). Semantic search. In Proceedings of the 12th international conference on World Wide Web (pp. 700-709).

[11] He, P., Gao, J., & Chen, W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv:2111.09543.

[12] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[13] Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., ... & Huang, X. (2023). A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. arXiv preprint arXiv:2303.10420.

[14] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.