Instructing Generative Models for the Task of Arabic Question Answering

Ahmed Hesham
Department of NLP
MBZUAI, Abu Dhabi, UAE
ahmed.aboeitta@mbzuai.ac.ae

Youssef Nafea
Department of NLP
MBZUAI, Abu Dhabi, UAE
youssef.nafea@mbzuai.ac.ae

Zain Muhammad Mujahid
Department of NLP
MBZUAI, Abu Dhabi, UAE
zain.mujahid@mbzuai.ac.ae

Abstract

This course project focuses on evaluating different models for question answering in Arabic, specifically using AraGPT2 and mGPT as the models for evaluation. The project employs manual evaluation, following a similar approach as the self-instruct paper, to rank the quality of the models' output. The evaluation aims to assess the performance of AraGPT2 and mGPT in generating accurate and contextually relevant answers to questions in Arabic. The results of the evaluation will provide insights into the strengths and weaknesses of these models for question answering in Arabic, and contribute to the body of knowledge in natural language processing and Arabic language processing.

1 Introduction

Question answering (QA) is a critical NLP task that involves generating accurate and informative answers to questions posed in natural language. With the increasing availability of large-scale language models, such as AraGPT2 (Antoun et al., 2021a) and mGPT (Shliazhko et al., 2022), there is growing interest in evaluating their performance for QA tasks in Arabic. In this course project, we aim to evaluate the performance of AraGPT2 and mGPT for question answering in Arabic using the ARCD dataset, which is an Arabic reading comprehension dataset specifically designed for QA tasks.

To improve the performance of these models in generating answers to questions, we will employ the Retrieval Augmentation Generation (RAG) technique (Lewis et al., 2020c). This approach combines the strengths of both retrieval-based and generative QA methods. RAG provides a more effective and robust solution for open-domain QA tasks by intelligently selecting and utilizing relevant context from large-scale knowledge sources. By integrating retrieval and generation components in a unified framework, RAG has shown promising results in various QA benchmarks and has the potential to significantly advance the development of state-of-the-art question answering systems.

In the context of RAG, a retrieval component, such as Dense Passage Retriever (DPR), identifies relevant passages from a large corpus of text that could help answer a given question. The generative component then uses the retrieved passages to generate a coherent and accurate answer. Dense Passage Retriever (DPR) (Karpukhin et al., 2020) is a retrieval system specifically designed for open-domain QA tasks. It employs a dual-encoder architecture that independently encodes the question and passage using deep learning models, such as BERT or RoBERTa, and calculates the similarity between the question and passage embeddings to identify the most relevant passages. In this project, we will integrate DPR with AraGPT2 and mGPT to create a RAG model for question answering in Arabic.

The ARCD dataset is a widely used benchmark for evaluating QA systems in Arabic. It contains a diverse range of questions that require reasoning, comprehension, and context awareness to generate accurate answers. The questions in the ARCD dataset cover various topics, including history, science, literature, and more, making it a comprehensive benchmark for evaluating the performance of AraGPT2 and mGPT in handling different domains and subjects in the Arabic language. To further enhance the training data for AraGPT2 and mGPT, we plan to use the NLLB (Team et al., 2022) model from Facebook to translate existing English QA datasets into Arabic due to its high performance in translating from English to Arabic, as it scored 48.3 sBLEU on FLORES-200 devtest (Team et al., 2022). This approach allows us to increase the training data for our models allows for wider coverage of the questions the model will be able to handle. We will employ manual evaluation, following a similar approach as the self-instruct paper, to assess the quality of the output generated by AraGPT2 and mGPT for question answering in Arabic. We will evaluate the models based on relevance using four classes of quality as in the self-instruct paper by performing human evaluation using two Arabic-speaking annotators on the generated answers. The results of the evaluation will provide valuable insights into the strengths and weaknesses of AraGPT2 and mGPT for QA in Arabic, and shed light on their potential for practical applications in Arabic language-related tasks.

This project contributes to the field of NLP and Arabic language processing by evaluating the performance of AraGPT2 and mGPT for question answering in Arabic using the ARCD dataset, and exploring the potential of leveraging translated English QA datasets to enhance the training data for these models. The findings of this evaluation will help in advancing the understanding of the capabilities and limitations of AraGPT2 and mGPT for QA in Arabic, and provide valuable insights for future research and development of QA systems in the Arabic language.

2 Related work

2.1 QA Datasets

English question answering datasets have been widely used for developing and evaluating question answering models. NarrativeQA (Kočiský et al., 2017) provides questions and answers based on long documents, such as books and movie scripts, making it suitable for evaluating models' ability to comprehend narratives. QnA (Joshi et al., 2017) dataset consists of questions from trivia quizzes and web searches, making it a diverse source of questions covering various topics. SQuAD (Rajpurkar et al., 2016) (Stanford Question Answering Dataset) is a widely used benchmark dataset that contains questions based on paragraphs from Wikipedia articles, making it a valuable resource for training and evaluating question answering models. TWEETQA (Xiong et al., 2019) focuses on questions from Twitter, making it suitable for evaluating models' ability to handle short and informal language. QuAC (Choi et al., 2018) (Question Answering in Context) provides questions based on passages from Wikipedia articles, with answers requiring reasoning over multiple sentences, making it suitable for evaluating complex question answering abilities. CoQA (Reddy et al., 2019) (Conversational Question Answering) contains dialogues between a user and an AI assistant, making it suitable for evaluating models' ability to engage in conversational question answering. HotpotQA (Yang et al., 2018) focuses on questions requiring information from multiple paragraphs for answering, making it suitable for evaluating models' ability to perform multi-hop reasoning. NQ (Kwiatkowski et al., 2019) (Natural Questions) contains questions from Google searches, making it suitable for evaluating models' ability to answer real-world questions. MultiSpanQA (Li et al., 2022a) introduces questions that require selecting multiple spans as answers, making it suitable for evaluating models' ability to perform complex span selection. MMCoQA (Li et al., 2022b) focuses on medical questions, making it suitable for evaluating models' ability to handle domain-specific question answering. TruthfulQA (Lin et al., 2022) focuses on questions that require identifying the correct and truthful answer among plausible but incorrect options, making it suitable for evaluating models' ability to handle deceptive questions. QuALITY (Pang et al., 2022) focuses on questions that require selecting the most informative answer among multiple plausible answers, making it suitable for evaluating models' ability to perform answer selection. ConditionalQA (Sun et al., 2021) introduces questions that require understanding and reasoning about hypothetical conditions, making it suitable for evaluating models' ability to handle conditional questions. CommonsenseQA (Talmor et al., 2019) focuses on questions that require common sense reasoning, making it suitable for evaluating models' ability to perform commonsense question answering. ArchivalQA (Wang et al., 2022) focuses on questions that require reasoning over historical documents, making it suitable for evaluating models' ability to perform historical question answering. QAConv (Wu et al., 2022) introduces questions that require reasoning over a conversation history, making it suitable for evaluating models' ability to perform conversational question answering. FairytaleQA (Xu et al., 2022) focuses on questions based on fairy tales, making it suitable for evaluating models' ability to comprehend narratives in a fantasy setting.

Arabic question answering datasets are not as common as English datasets, but they are valuable resources for evaluating question answering models on Arabic language understanding and reasoning abilities. Arabic-SQuAD (Mozannar et al., 2019) is an Arabic translated version of the widely used SQuAD dataset, consisting of questions based on Arabic Wikipedia articles, making it suitable for evaluating models' ability to answer questions in Arabic based on textual context. ARCD dataset (Mozannar et al., 2019) (Arabic Reading Comprehension Dataset) focuses on questions from Arabic news articles, making it suitable for evaluating models' ability to comprehend news-related texts and provide accurate answers.

Multilingual question answering datasets are important for evaluating models' ability to perform question answering in multiple languages. All the following datasets are multilingual and contain Arabic samples. MLQA (Lewis et al., 2020b)(Multilingual Question Answering) is a benchmark dataset that consists of questions in various languages, making it suitable for evaluating models' ability to perform question answering in multiple languages. XQuAD (Artetxe et al., 2019) (Cross-lingual Question Answering Dataset) is another multilingual dataset that provides questions in English and their corresponding passages in various languages, making it suitable for evaluating models' ability to perform cross-lingual question answering. XQA (Liu et al., 2019) (Cross-lingual Question Answering) dataset focuses on questions and answers in multiple languages, making it suitable for evaluating models' ability to handle multilingual question answering. TydiQA (Clark et al., 2020) is a multilingual dataset that covers a wide range of languages, making it suitable for evaluating models' ability to perform question answering in diverse languages. XOR QA (Asai et al., 2021) (Cross-lingual Open-Reading Question Answering) focuses on questions based on Wikipedia articles in multiple languages, making it suitable for evaluating models' ability to perform cross-lingual question answering with open-domain information. XQuAD-R (XQuAD with Reranker) (Roy et al., 2020) is an extension of XQuAD dataset with additional passages for re-ranking, making it suitable for evaluating models' ability to perform passage ranking for cross-lingual question answering. MKQA (Longpre et al., 2021) (Multilingual Knowledge-based Question Answering) is a multilingual dataset that requires models to answer questions by leveraging knowledge from external knowledge sources, making it suitable for evaluating models' ability to perform knowledge-based question answering in multiple languages.

In summary, question answering datasets in English, Arabic, and multilingual settings provide diverse sources of questions, covering various topics, domains, and languages. A summary of the datasets is available in table 1. These datasets are crucial for training and evaluating question answering models and advancing the field of natural language processing. They enable researchers to develop models that can comprehend and reason over different languages, domains, and types of textual context, which is essential for building robust and versatile question answering systems for real-world applications.

2.2 Retreival Augmentation Techniques

Retrieval augmentation techniques involve incorporating external knowledge sources to enhance the performance of QA models. In recent years, various methods have been developed to improve document retrieval in question-answering tasks.

(Karpukhin et al., 2020) proposed a dense retrieval approach called Dense Retriever, which uses dense vector representations to retrieve passages relevant to the input questions. This method employs bi-encoder architecture with a BERT-based model for both questions and passages, enabling the system to learn dense vector representations that can be used to retrieve relevant documents efficiently, which will help the model answer questions by providing context. The study demonstrated that the dense retrieval approach significantly outperforms traditional sparse retrieval methods, such as BM25 (Robertson, 2009), in open-domain QA tasks.

(Lee et al., 2019) proposed the REALM model, which combines pre-trained language models with a retrieval-augmented mechanism. The model retrieves and ranks relevant documents from a prebuilt knowledge corpus based on their semantic similarity to the input query. REALM demonstrated significant improvements in performance on open-domain QA tasks compared to previous state-of-the-art models.

Furthermore, Guu et al. (2020) (Lewis et al., 2020c) developed the Retrieval-Augmented Generation (RAG) model, which combines the strengths of pre-trained language models and retrieval-based methods. By integrating the BART

Dataset Name	Year	Language	Samples	Annotation	Context
SQuAD	2016	English	161k	Crowdworkers	Wiki
SearchQA	2017	English	140k	Google	Search Snippets
NewsQA	2017	English	100k+	Crowdworkers	News
NarrativeQA	2017	English	46.8k	Crowdworkers	Wiki
QnA	2018	English	1m pairs	Human editors	Web
QuAC	2018	English	98k	Crowdworkers	Wiki
CoQA	2018	English	127k	Crowdworkers	Mixed
HotpotQA	2018	English	113k	Crowdworkers	Wiki
NQ	2019	English	323k	Human Editors	Wiki
TWEETQA	2019	English	13.7k	crowdworkers	Tweets
CommonsenseQA	2019	English	12k	Crowdwrokers	Wiki
ConditionalQA	2021	English	3.1k	Crowdworkers	UK Gov. Website
MultiSpanQA	2022	English	19k	Crowdworkers	Wiki
MMCoQA	2022	English	6k	Human Editors	Wiki
TruthfulQA	2022	English	817	Human Editors	Hand-Crafted
QuALITY	2022	English	6.7k	Crowdworkers	Long Articles
ArchivalQA	2022	English	532k	Crowdworkers	Wiki
QAConv	2022	English	35k	Crowdworkers	Conversations
FairytaleQA	2022	English	10k	Human Editors	Stories
Arabic-SQuAD	2016	Arabic	48k	Machine Translation	Wiki
ARCD	2019	Arabic	1.4k	Crowdworkers	Wiki
MLQA	2019	Mixed	36k across 8 languages	Crowdworkers	Wiki
XQuAD	2019	Mixed	12k across 10 languages	Translators	Wiki
TyDiQA	2020	Multi	200k across 11 languages	Human Editors	Wiki
XQuAD-R	2020	Mixed	12k across 10 languages	Translators	Wiki
MKQA	2021	Multi	260k across 26 language	Crowdworkers	Wiki
XOR QA	2021	Mixed	40k across 7 languages	Machine Translation	Wiki

Table 1: Sample of previous dataset benchmarks with the number of questions in each dataset, how annotation was done, and the context of the data. The datasets are grouped by language

model and the Dense Retriever, RAG demonstrated its effectiveness in various QA settings, including factoid and non-factoid questions, where factoid questions have simple facts as answers, whereas non-factoid questions typically have as answers longer pieces of readable information which may come from single or multiple documents. The model's performance surpassed that of other retrieval-based and generation-based models on multiple benchmarks.

Knowledge graphs have emerged as a promising approach to improve document retrieval in question-answering tasks. A notable advancement in this area is the work of Sun et al. (2018), (Tu et al., 2019) who proposed a method called Multi-hop Reading Comprehension across Multiple Documents (MRC-MD) for retrieving information from knowledge graphs. MRC-MD leverages an iterative reasoning process to identify relevant entities and relations in the knowledge graph, enabling the model to answer complex multi-hop questions. The study demonstrated that MRC-MD outperforms existing methods in terms of accuracy and reasoning capability on QA tasks involving knowledge graphs.

Another significant contribution to the field is the work of Xiong et al. (2019), (Zhang and Qian, 2020) who developed the Graph Convolution over Pruned Dependency Trees (GraphDP) model. GraphDP incorporates syntactic and semantic information from dependency trees, in addition to graph convolutional networks (GCNs), to better capture and represent the relationships between entities in the knowledge graph. The model showed substantial improvements in performance on the SQuAD benchmark dataset, surpassing other state-of-the-art models that do not employ knowledge graphs.

Additionally, Vashishth et al. (2020) (Ren et al., 2020) proposed the Query2Box model, which leverages hyperbolic embeddings to represent hierarchical structures in knowledge graphs. Query2Box can efficiently retrieve relevant information from knowledge graphs by formulating the QA task as a set intersection problem. The model demonstrated its effectiveness in various QA settings, including single and multi-hop queries, as well as entity and relation extraction tasks.

2.3 QA Approaches

QA systems allow users to search for information in different formats, including unstructured and structured data, using natural language. This is an important aspect of conversational AI, which includes conversational question answering (CQA) as a specialized research topic. In CQA, the system needs to comprehend the context and engage in multiple rounds of QA to fulfill the user's information requirements.

2.3.1 QA over KB

Knowledge-based (KB) question answering using a structured knowledge base has many advantages, such as accurate and reliable answers, support for complex queries, and applicability to various domains. However, knowledge bases can be time-consuming and expensive to create and maintain, and they may suffer from incompleteness or inaccuracies. Additionally, knowledge-based QA may struggle with questions that require inferencing or reasoning beyond what is explicitly stated in the knowledge base.

Question answering over large-scale knowledge-based systems has evolved from simple single-fact tasks to more complex queries that require multi-hop interaction and traversal of knowledge graphs. These queries fall under the category of single-turn QA, where a user asks a question and the system finds the best answer (Guo et al., 2018). While KB-QA systems have significantly improved the flexibility of the QA process, it's unrealistic to expect these systems to handle complex queries without a complete understanding of the KB's organizational structure. Therefore, a sequential KB-QA system is a better option, as it allows users to query the KB step-by-step.

Semantic parsing is a crucial technical aspect in the development of KB-QA systems, as it involves converting natural language text into meaningful logical forms (Cheng et al., 2019). Once a correct logical form is obtained, it can be executed as a query on the knowledge source to retrieve answer denotations. In other words, semantic parsing allows for the mapping of natural language into logical forms, which are then used to query the knowledge source and obtain answers.

(Iyyer et al., 2017) introduced the concept of semantic parsing for sequential QA by developing a dataset of simple interrelated questions derived from a complicated WikiTableQuestions dataset (Pasupat and Liang, 2015). They proposed a model called Dynamic Neural Semantic Parsing (DynSP), which is a weakly supervised structured output learning approach based on reward-guided search. The model creates a semantic parsing problem as a state-action search problem, where each state denotes a partial or complete parse, and each action is an operation to extend the parse. Unlike traditional parsers, DynSP explores and constructs various neural network structures for different questions. By using DynSP, the model can effectively answer complex queries by understanding the underlying relationships between the questions and the provided table.

2.3.2 GPT-Based QA

One advantage of GPT-based models for question answering is their ability to generate text in response to questions, which can provide more natural and coherent answers than other types of models that rely on pre-defined answer choices or retrieval-based methods. GPT-based models can also handle more complex and open-ended questions, as they can generate a wide range of possible answers based on their training. However, there are also some challenges to using GPT-based models for question answering. One challenge is the potential for the model to generate incorrect or irrelevant answers, particularly if the training data is limited or biased. Another challenge is the potential for the model to generate overly long or convoluted answers that are difficult for users to understand. Also, they require a huge amount of training data to make the model generate acceptable outputs.

NLG-LM (Tu et al., 2019) is a framework that uses multitask learning to generate not only semantically correct responses but also maintain the naturalness of conversations. The model employs a sequence-to-sequence architecture to train two tasks simultaneously: natural language generation (NLG) and language modeling (LM). The decoder incorporates the language modeling task on human-generated utterances to enhance the language-related elements. Additionally, the unsupervised nature of the language model eliminates the need for an extensive amount of unlabeled data for training. With these features, NLG-LM has shown great potential in generating more natural and coherent responses for various NLP tasks.

Generating system utterances for a user is a crucial step in NLP tasks, and the use of pre-trained language models has revolutionized the field of language generation in recent years. A model developed by (Peng et al., 2020), called semantically conditioned generative pre-training (SC-GPT), is based on OpenAI's Generative Pre-training (GPT) (Radford et al., 2018). SC-GPT is designed to generate responses based on the semantic meaning of the input text. This model has shown great promise in improving the accuracy and relevance of generated responses.

Recent advancements in language models have facilitated the development of more advanced conversational AI models and dialogue agents that can answer questions while also engaging in conversations with users. ChatGPT, introduced by OpenAI, and LaMDA (Thoppilan et al., 2022), a family of transformer-based language models (Vaswani et al., 2017) for dialogue applications, are examples of these chatbots. Additionally, Sparrow (Glaese et al., 2022) is a dialogue agent trained using reinforcement learning with human feedback, while webGPT (Nakano et al., 2022) is a fine-tuned GPT-3 model that answers questions by allowing the language model to search the web. Meena is another chatbot that has been trained to respond in a human-like manner by providing sensible responses (Adiwardana et al., 2020).

2.3.3 QA for Conversation

Knowledge graph based methods can be useful for conversation-based question answering, as they provide a structured representation of knowledge that can be used to store and retrieve relevant information during the course of a conversation. By tracking the context of the conversation, the knowledge graph can be updated and expanded as new information is added, allowing for more comprehensive and accurate answers. However, conversation-based question answering can also be challenging for knowledge graph based methods, as it requires the ability to understand and follow the flow of the conversation, and to provide relevant and accurate information in real-time. This can require more advanced techniques such as natural language processing, dialogue management, and machine learning.

EDGQA, introduced by (Hu et al., 2021), utilizes the Stanford CoreNLP parser to generate a constituency parsing tree for the input question, which is then transformed into an entity description graph (EDG) using human-curated heuristic rules. The EDG is a root-acyclic graph that represents the entities and relationships in the question. EDGQA uses different linking methods, such as Falcon (Sakor et al., 2019), to link the nodes in the EDG to the corresponding entities in the target KG. However, these methods require building indices in a pre-processing phase before the retrieval can take place. In addition, EDGQA extracts all entity types from the KG in a pre-processing phase for filtration. Finally, the semantically equivalent SPARQL query is generated by mapping the linked vertices, predicates, and entity types to the different phrases in the input question.

The state-of-the-art for answering questions on knowledge graphs is KGQAN (Omar et al., 2023). This system uses a Seq2Seq pre-trained model such as BART (Lewis et al., 2020a) or GPT-3 to train a triple-patterns generation model, which is then converted into a graph structure called a Phrase Graph Pattern (PGP). KGQAN performs just-in-time linking based on built-in indices in the KG engines and utilizes FastText (Bojanowski et al., 2017) word embedding models to assess the semantic affinity between phrases in the question and vertices and predicates in the KG. The unique feature of KGQAN is that it doesn't rely on any pre-processing index or models trained on the KG to perform the linking, making it capable of answering questions on any knowledge graph as an on-demand service.

We adapt the GPT-Based QA approaches for our work on the arabic language by using existing generative arabic, and multilingual models because of the quality and uniqueness of the text it can generate. This work adapts the self-instruct methodology for arabic question answering using generative models.

3 Dataset

In this work, we're currently using ARCD (Arabic Reading Comprehension Dataset), a publicly available dataset, as our dataset which consists of approximately 1400 samples containing context, question, and answer in arabic. The dataset contains paragraphs from Wikipidea, and the questions asked are often simple, requiring short answer. Additionally, we use Arabic-SQuAD to increase the coverage of the model, knowing that the data won't have the same quality given that it's machine translated. In our work we train on all of the data, while leaving 30 samples for manual evaluation. In the future, we're planning to add more datasets as we will use Facebook's model NLLB (No language left behind) to translate some of the already existing English datasets to Arabic to further increase the training set size, we also plan to increase the amount of evaluation data to 200 samples.

4 Models

Aragpt2 and mGPT are two state-of-the-art language models developed by OpenAI that have significantly advanced the field of natural language processing (NLP). Aragpt2, also known as Arabic GPT-2, is a variant of the GPT-2 model that is specifically designed for the Arabic language. It has been fine-tuned on 77GB Arabic text, making it highly capable of generating coherent and contextually relevant Arabic language text. Aragpt2 has been widely used for a variety of NLP tasks, including text generation, language translation, and sentiment analysis (Antoun et al., 2021b), and has demonstrated impressive performance in these tasks, showcasing its potential for numerous applications in Arabic NLP. The model is available in 4 different settings, with the largest one having 1.46B parameters.

On the other hand, mGPT, or Multilingual GPT, is a variant of the GPT model that has been trained on multiple languages, making it capable of handling text in multiple languages, including but not limited to English, Spanish, French, Chinese, and more. mGPT is trained on an extensive multilingual corpus of around 500B UTF characters, enabling it to capture the nuances of different languages and perform well in various cross-lingual tasks, such as cross-lingual text classification, entity recognition, and machine translation (Shliazhko et al., 2022). The versatility and adaptability of mGPT make it a valuable tool for multilingual NLP research and applications, where it can be leveraged to process and generate text in multiple languages with high accuracy and fluency. The model is available in 2 different settings, with the largest one having 13B parameters.

We use Aragpt2-large which has 792M parameter, Aragbt2-base which has 153M parameter, and the mGPT version with 1.3B parameters from huggingface. we compare the difference in size of the model, as well as comparing a multilingual model with arabic models. we train the small model for 30 epochs, while we trained the larger models for only 5 epochs each, we picked the number of epochs by trial and error.

5 Methodology

In this section, we outline the methodology employed for implementing the Retrieval Augmentation Generation (RAG) model for question answering in Arabic, incorporating Dense Passage Retriever (DPR) for passage retrieval. Both the generator and retriever components will be fine-tuned on our training data to adapt to the specific characteristics and nuances of the Arabic language. The methodology consists of the following steps: data preparation, passage retrieval using DPR, RAG model implementation and model finetuning.

5.1 Data Preparation

To prepare the data for the fine-tuning process, we process each example in the dataset to follow the format: paragraph + \n + question + \n + answer + EOS token, where EOS (End of Sentence) token is included to signal the model the end of the generated answer, ensuring that the model focuses on generating a single coherent answer and does not continue generating irrelevant information. This format helps the model understand the structure of the input and output during the fine-tuning process.

5.2 Passage Retrieval using Dense Passage Retriever (DPR)

Dense Passage Retriever (DPR) (Karpukhin et al., 2020) is employed as the retrieval component in the RAG pipeline for identifying relevant passages that can help answer the given questions. To implement DPR, we follow the steps outlined below:

1. **Fine-tuning the Passage and Question Encoders:** Both the passage and question encoders are fine-tuned on our training data to adapt to the Arabic language. The fine-tuning process consists of training the encoders to generate dense vector representations that maximize the similarity between correct question-answer pairs while minimizing the similarity between incorrect pairs.

2. **Indexing:** The entire Arabic corpus, including the ARCD dataset and the translated QA dataset, is indexed using the fine-tuned DPR's passage encoder. Each passage is encoded into a dense vector representation, which is stored in a large-scale vector database, such as FAISS (Johnson et al., 2019).

3. **Retrieval:** The similarity between the encoded questions and indexed passages is calcu-

lated using FAISS. FAISS (Facebook AI Similarity Search) is an efficient similarity search library that allows for quick and accurate retrieval of the top-k most similar vectors, in our case, relevant passages. It leverages an approximate nearest neighbor search algorithm to efficiently handle high-dimensional data, which is crucial for the large-scale retrieval tasks in the RAG framework. The top-k most similar passages are retrieved as context for generating answers.

5.3 RAG Model Implementation

The RAG model implementation consists of integrating the AraGPT2 or mGPT models with the retrieval component (DPR) to create a unified framework for question answering in Arabic. The following steps are performed:

1. **Context Fusion:** The top-k retrieved passages from DPR are concatenated to form a single context input. If total context length exceeds the maximum token limit of the language models, we just truncate.

2. **Conditional Generation and Fine-tuning:** The language model (AraGPT2 or mGPT) is fine-tuned on the prepared dataset, conditioning on the provided context. During the fine-tuning process, the model learns to generate answers based on the given question and the context retrieved by the DPR. The model is optimized using standard optimization techniques, such as stochastic gradient descent or Adam optimizer.

6 Experiments

We follow the same approach for fine-tuning as in the self-instruct paper to fine-tune our models, using ARCD dataset instead of the LM generated data, then we prompt the models on the evaluation set and manually classify the results into one of four classes from A to D based on the quality of the output with A being accurate and correct answers, and D being hallucinations from the model. We compare the results of the manual classification between the two models and provide further analysis in the evaluation section. Furthermore, we select the best performing model and fine-tune it on the whole dataset (ARCD + Arabic-SQuAD) except for 30 samples which are used to apply the same style of evaluation.

For the augmented retrieval module, we train it on the whole dataset as well, except for 30 samples for evaluation, and we evaluate it by classifying the top 10 output contexts into one of three classes:

- **Hit:** At least one of the top 10 contexts provided is relevant and contains the answer of the provided question

- **Relevant:** At least one of the top 10 contexts provided is relevant, but doesn't include the answer of the question

- **Irrelevant:** Non of the contexts provided is relevant to the question

7 Evaluation

For evaluating the fine-tuning process, a small evaluation set of 8 prompts is constructed where each prompt contains a paragraph and a question on the context of this paragraph and the generated answer is manually evaluated by two native arabic speakers in the same fashion as in the self-instruct paper. The evaluation set was chosen without replacement from the dataset we are using. outputs of the models are available upon request.

A quite noticeable differences in the generated answers between the mgpt model and the aragpt2 model (either the base or the large one).

For the pre-trained mGPT model performed poorly on the evaluation set, the following are examples of mistakes in the output answer:

1. It always hallucinate.

2. It always generate questions of its own and answer those question

3. Some times it qives the right answer but it will among the hallucinations and you need to carefully search for it

For the pre-trained aragpt-base model, it performed much better than the mgpt one, it was able to give answers and avoid hallucinations, the following are examples of mistakes in the output answer:

1. It rarely hallucinate.

2. Most of the times it answers the wrong answer.

3. Sometimes it gives a irrelevant answers (answers to a different question).

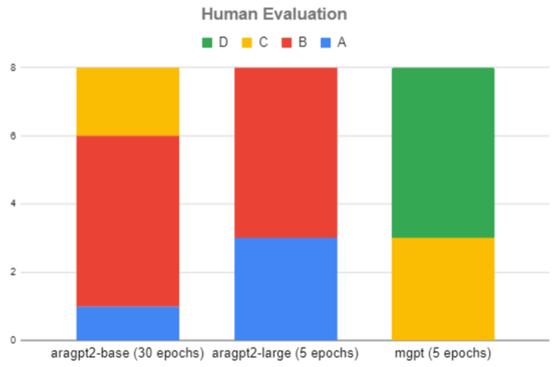

Figure 1: Human evaluation results for the three models, where each output instance is assigned a letter based on quality. A is the label for higher quality outputs while D is the lowest.

For the pre-trained aragpt-large model, it gave the best performance among the others,where only finetunied on 5 epochs, nearly half of the answers were perfect, and other answers were just wrong answer to the question, it was able to avoid any kind of hallucinations, in addition to giving irrelevant answers.

The results of the manual evaluation is shown in Figure 1 which shows the difference between the quality of output of each of the three models. What's interesting in the results is that mGPTQ which is the largest of the models performs poorly as it hallucinates most of the time, and other times it would give the correct answer, but continue generating more information, even information from outside of the context, this is possibly because mGPT has data from Arabic Wikipedia in its training data which could be a reason on to why it generates out of context answers. On the other side, we notice that both aragpt2 models perform well on the task, with a slight edge to the large model which is expected as it's almost 6 times bigger than the base model. These results could imply that the amount of training data wasn't enough for the mGPT model to adapt to the task.

Additionally, the best model was trained on the bigger training set containing ARCD, and Arabic-SQuAD with 30 samples for manual evaluation (were randomly selected so most of them were from Arabic-SQuAD). The results of this experiment is shown in Figure 2. The results shows that the model's performance hugely declined as it hallucinated almost in half of the questions, while only 20% of the answers were considered acceptable. This was because the translation of the SQuAD dataset wasn't a "clean" translation, as many named entities were still in English, or sometimes would be translated in one paragraph but not in other paragraphs, also the translation is not always contextually correct, all of which would make the model's job more difficult in answering the questions. Furthermore, having most of the evaluation-set being from Arabic-SQuAD wasn't optimal as the dataset doesn't represent a real-worlds scenarios given that the dataset is not cleanly translated.

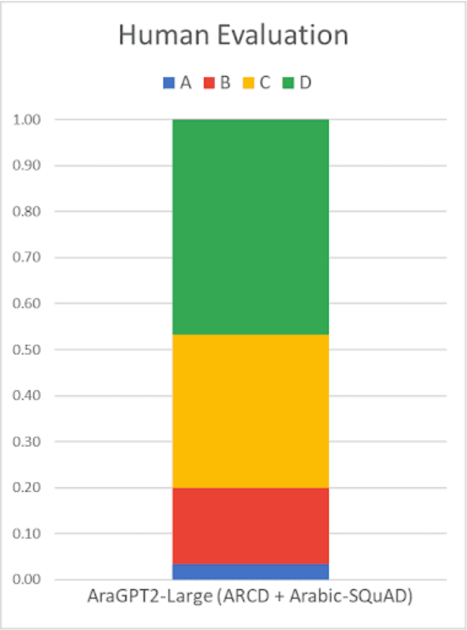

Figure 2: Human evaluation results for AraGPT2 on ARCD + Arabic-SQuAD, where each output instance is assigned a letter based on quality. A is the label for higher quality outputs while D is the lowest.

The results for the human evaluation experiment for the augmented retrieval module can be found in Figure 3. The figure shows that in 60% of the samples, the top 10 contexts provided by the module weren't even relevant to the topic of the question, while the other 40% had relevant contexts discussing the same topic, but didn't include the answer to the provided question. This is possibly because similar to the previous experiment, most of the manual evaluation samples were from Arabic-SQuAD, and upon investigating the samples we found that most of the questions weren't complete sentences and assumed that the context was already given. Furthermore the augmented retrieval module was only selecting from the contexts in the training set, which we plan on solving by making the module select contexts straight from Wikipedia as we believe this will make the module cover a

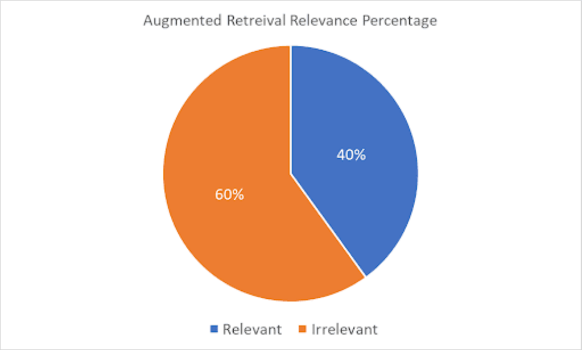

Figure 3: Human evaluation results for the augmented retreival module on ARCD + Arabic-SQuAD, where each output instance is assigned to one of the three classes: Hit, Relevant, and Irrelevant.

wider range of contexts.

8 Conclusion and Future Work

In this report, our focus was to conduct an extensive literature review on the various techniques used in our project. Additionally, we conducted some experimentation using ARCD and Arabic-SQuAD datasets, to fine-tune three models: two AraGPT models and one GPT. While our initial experiments were limited in scope, they allowed us to gain valuable insights into the performance of our models and helped us identify areas that require further improvement.

Moving forward, our plan is to increase the number of datasets we use and the number of manually evaluated samples by using state-of-the-art machine translation models to have clean translations of existing English QA datasets. Furthermore, we intend to improve our augmented retrieval module to make the conversational AI more robust. We believe that this step will support our QA model greatly by providing it contexts it could use to answer questions, thus improving the overall user experience. Ultimately, our aim is to develop a conversational AI that can seamlessly engage with users and provide them with relevant and accurate information, making it an indispensable tool for a variety of applications.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021a. Aragpt2: Pre-trained transformer for arabic language generation.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021b. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2019. Learning an Executable Neural Semantic Parser. *Computational Linguistics*, 45(1):59–94.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac : Question answering in context.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages.

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. *Advances in Neural Information Processing Systems*, 31.

Xixin Hu, Yiheng Shu, Xiang Huang, and Yuzhong Qu. 2021. Edg-based question decomposition for complex question answering over knowledge bases. In *The Semantic Web – ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings*, page 128–145, Berlin, Heidelberg. Springer-Verlag.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The narrativeqa reading comprehension challenge.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022a. MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022b. MMCoQA: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231, Dublin, Ireland. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019. XQA: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, Florence, Italy. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. 2019. Neural Arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118, Florence, Italy. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback.

Reham Omar, Ishika Dhall, Panos Kalnis, and Essam Mansour. 2023. A universal question-answering platform for knowledge graphs.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

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

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. *arXiv preprint arXiv:2002.05969*.

S. Robertson. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. Lareqa: Language-agnostic answer retrieval from a multilingual pool. *arXiv preprint arXiv:2004.05484*.

Ahmad Sakor, Isaiah Onando Mulang', Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019. Old is gold: Linguistic driven approach for entity and relation linking of short text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2336–2346, Minneapolis, Minnesota. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2021. Conditionalqa: A complex reading comprehension dataset with conditional answers.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. *arXiv preprint arXiv:1905.07374*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open domain question answering over historical news collections.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2022. Qaconv: Question answering on informative conversations.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: Fairytaleqa – an authentic dataset for narrative comprehension.

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.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 3540–3549.