Efficient Fine-tuning Large Language Models for Knowledge-Aware Response Planning

Minh Nguyen[1⋆], Kishan K C[2], Toan Nguyen[2], Ankit Chadha[2], and Thuy Vu[2]✉

[1] Department of Computer Science, University of Oregon, OR, USA
[2] Amazon Alexa AI, CA, USA
minhnv@cs.uoregon.edu
{ckshan,amztoan,ankitrc,thuyvu}@amazon.com

Abstract. Large Language Models (LLMs) have shown impressive emergent language capabilities, especially in applications with high ambiguity, such as language reasoning and knowledge consolidation. However, previous work explores the use of LLMs for acquiring information using either parametric or external knowledge, which might lead to serious issues such as hallucination. Toward solving these issues, we present a novel approach of knowledge-aware response planning (KARP) and propose a novel framework that employs (i) a knowledge retriever to obtain relevant information from web documents or databases for a given user query, and (ii) a robust fine-tuning strategy for LLMs to exploit the retrieved external knowledge for planning a final response. Experimental results show that our proposed framework can provide natural, concise answers for open-domain questions with high accuracy.

Keywords: Knowledge-Aware Response Planning · Question Answering · Large Language Models · Fine-tuning.

1 Introduction

General question answering (QA), a crucial natural language processing (NLP) task, is often regarded as AI-complete [62, 8]; that is, QA will only be considered solved once all the challenging problems in artificial intelligence (AI) have been addressed. Several virtual response assistants, including Google Assistant, Amazon Alexa, and Apple's Siri, have integrated state-of-the-art QA technologies, allowing them to understand and generate responses in natural languages, providing valuable services to users. However, general QA still presents significant challenges, primarily due to the inherent difficulties in reasoning with natural language, including aspects like commonsense and general knowledge. Past research has explored the use of Large Language Models (LLMs) for general QA, predominantly leveraging either parametric (e.g., ChatGPT[3]) or external (e.g., WebGPT[38]) knowledge sources. This method, however, can lead to considerable complications, including hallucination - the generation of plausible but incorrect or unverified information. To address these challenges, this paper introduces the concept

⋆ This work was completed while the author was an intern at Amazon Alexa AI.
[3] https://chat.openai.com/chat

q:	What college offers chiropractic ?
c_1:	New York Chiropractic College offers 1 Chiropractic Degree program. It's a private university in a far away town. In 2015, 173 students graduated in the study area of Chiropractic with students earning 173 Doctoral degrees.
a_1:	New York Chiropractic college offers chiropractic.
c_2:	Chiropractic care is also essential for college students who want to stay healthy. The central nervous system is based in the spinal column, so correcting subluxations (misalignments) of the spine is important, no matter how old you are. Holt Chiropractic in Port Orchard, WA provides expert chiropractic care to students of all ages.
a_2:	Holt Chiropractic College offers chiropractic.
c_3:	Howell Township is a township in Monmouth County, New Jersey, United States. As of the 2010 United States Census, the township's population was 51,075, reflecting an increase of 2,172 from the 48,903 counted in the 2000 Census.
a_3:	Howell Township College offers chiropractic.

Table 1: Generated answers for a question q with different context passages c_1 (relevant), c_2 (quasi-relevant), and c_3 (irrelevant) from MS MARCO QA NLG test set [39]. Answers a_1, a_2, and a_3 are generated by GenQA [17].

of Knowledge-Aware Response Planning (KARP) for general QA along with a novel framework that combines a knowledge retriever with a robust fine-tuning strategy for LLMs. In particular, the problem of KARP can be defined as follows. Given a user query and a prompt containing external knowledge, the goal is to develop a model that can consolidate a response that must be crafted not just from the externally sourced information, but also from the model's inherent parametric knowledge. This is different from the previous work that aim to generate a response by either harnessing parametric knowledge (e.g., ChatGPT) or retrieving from external knowledge such as knowledge bases [2, 3, 65, 51], web documents [68, 6, 67, 7, 13], or a provided context [15, 45, 59, 10].

With the emergent abilities of LLMs [61], generative QA systems, in which answers are produced by a generative LLM, have been explored to improve the performance of QA [21, 49, 43, 19, 27, 18, 12, 37]. In paritcular, previous work typically employs pre-trained LLMs with encoder-decoder architectures such as BART [28] and T5 [44], where the encoder consumes a given question and a *required* relevant context as input for the decoder to generate an answer to the question [24, 17]. On one hand, the similarity between generative QA and the pre-training tasks of LLMs enables transfer learning to improve QA performance. On the other hand, the generative formulation allows for flexibility in handling various types of QA problems (e.g., extractive QA, multiple-choice QA) [24]. However, a well-known issue that has been shown to occur with the generative models is hallucination [35, 50, 53], where the models generate statements that are plausible looking but *factually* incorrect. Additionally, if the answers are composed by a pretrained LLM without external knowledge, i.e., using parametric knowledge, the information contained in the answers might be outdated and no longer valid. For example,

the answer for the question *"Which country is the reigning World Cup champion?"* will change through time.

Recent works such as GenQA [17] and WebGPT [38] mitigate these issues by employing an information retrieval component, which is responsible for collecting web content to compose an answer for a given question. Formally, given a question q and a retrieved web content c, the model is trained to take (q, c) as input to produce a response $a = f_\theta(q, c)$, where f_θ denotes the corresponding LLM with the parameters θ. Unfortunately, f_θ may merely learn to copy/synthesize information from c to produce a if c often contains necessary information for correctly answering the question q in training data such as MS MARCO QA NLG [39][4]. As a result, the model may fail to provide a correct answer for a given question if the retrieved content is missing or contains (quasi-)irrelevant information (see Table 1). In other words, performance of these retrieval-based QA models are limited to an upper bound by the knowledge retriever.

In this work, we address such issues in building a generative QA model. First, we utilize a knowledge retriever that employs Optimal Transport to selectively identify relevant content from web documents or databases for a given user query. Second, we propose a novel fine-tuning strategy specially designed for LLMs, which combines external knowledge, i.e., provided by the knowledge retriever and the intrinsic pre-trained knowledge in LLMs, wherever possible, to generate informed responses.

Particularly, we propose the knowledge retriever as a dense passage retriever (DPR) model. Our proposed DPR model performs an alignment between a given question and a text passage via Optimal Transport to find relevant information in the passage for determining its correctness. The relevant context in the passage will then be used to produce a correctness score for ranking. In this way, we can obtain top k text passages from databases/web documents, which are treated as external knowledge in our framework. Different from GenQA and WebGPT that follows a single-style "$a = f_\theta(q, c)$" finetuning strategy, we propose to employ a multi-style finetuning strategy, where both "$a = f_\theta(q, c)$" and "$a = f_\theta(q)$" are used to train the model. The latter intentionally excludes the external knowledge c from the input to encourage the model to retrieve its own knowledge from the model parameters θ, which have been pretrained on massive unlabeled text data [28, 44, 11, 56]. To combine the two finetuning styles, we propose to finetune the LLM with "$a = f_\theta(q, c)$", and sequentially finetune the model with "$a = f_\theta(q)$". At test time, we use the "$a = f_\theta(q, c)$" style to make predictions. Experimental results show that our proposed finetuning strategy significantly improves the performance compared to the baselines on MS MARCO QA NLG, demonstratting the effectiveness of our proposed method. Finally, we scale up our framework to further improve the QA performance by training the model i) with "$a = f_\theta(q, c)$" on QA datasets such as SQUAD [45] (c is a context passage), MCTest [48] (c consists of multiple choices), Anthropic [1] (c is the previous question in a conversation), and ii) with "$a = f_\theta(q)$" on QA datasets such as WikiQA [69] and Wdrass [73].

[4] All answers a in MS MARCO QA NLG are written by human annotators based on summarizing answer information in context passages c.

Our experiments show that the resulting system behaves as a knowledge aware response planner that provides natural, concise answers for open-domain questions with high accuracy.

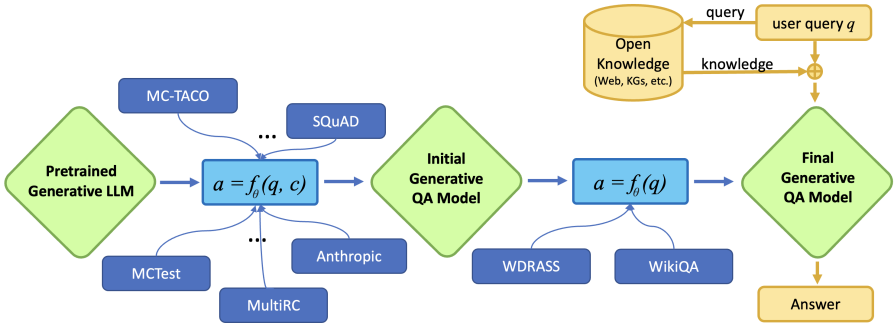

Fig. 1: Overview of our proposed framework for KARP. The blue and orange arrows represent the finetuning and inference processes of our model respectively.

2 Proposed Method

2.1 Knowledge-Aware Response Planning

The problem of Knowledge-Aware Response Planning (KARP) can be outlined as follows: Given a user query and a prompt loaded with external knowledge, the aim is to build a model capable of formulating a response. This response should be planned not only from the external information provided but also derived from the model's inherent parametric knowledge.

To this end, our proposed framework for KARP consists of (i) a knowledge retriever and (ii) a generative LLM-based question answering model. An overview of our framework is shown in Figure 1. Details regarding the knowledge retriever and the generative QA model are presented in section 2.2 and 2.3, respectively.

2.2 Knowledge Retriever

Our knowledge retriever functions as a dense passage retrieval (DPR) system. Given a question q and a group of N text passages $C = \{c_1, c_2, \ldots, c_N\}$, the goal of DPR is to determine the correct answer passages $A \subset C$ by learning a reranking function $r : Q \times \phi(C) \rightarrow \phi(C)$, where Q represents the set of questions and $\phi(C)$ represents all the possible orderings of C. The intent is to place the answer passages A at the top of the ranking produced by the function r. The reranker r is typically a pointwise network $f(q, c_i)$, such as TANDA [13], which learns to assign a correctness score $p_i \in (0, 1)$ to each text passage c_i for ranking purposes. Our focus lies on the contextual DPR, where

supplementary context, like surrounding context, is used to more accurately ascertain the validity score of an answer passage.

Our knowledge retriever consists of three primary elements: i) Encoding, ii) Question-Context Alignment with Optimal Transport (OT), and iii) Answer-Context Dependencies. The diagram of our suggested model can be seen in Figure 2.

Encoding We are provided with a question represented as $q = [w_1^q, w_2^q, \ldots, w_{T_q}^q]$ with T_q words and a set of N text passages $C = \{c_1, c_2, \ldots, c_N\}$ retrieved from a search engine. Each passage, denoted as $c_i = [w_1^c, w_2^c, \ldots, w_{T_c}^c]$, consists of T_c words. In this work, we consider previous and next passages c_{prev}, c_{next} as additional context for each candidate passage $c \in C$. To create the input for our DPR model, we concatenate the question, answer passage, and context passages into a single input sequence: $[q; c; c_{prev}; c_{next}]$. This combined sequence is then passed through a pre-trained language model (PLM), e.g., RoBERTa [33], to obtain contextualized word embeddings. Additionally, we employ distinct segment embeddings for the question, answer passage, and context passages. These segment embeddings, which are randomly initialized and trainable during training, are added to the initial word embeddings in the first layer of the PLM. For simplicity, let $[\mathbf{w}_1^q, \mathbf{w}_2^q, \ldots, \mathbf{w}_{T_q}^q]$ and $[\mathbf{w}_1^c, \mathbf{w}_2^c, \ldots, \mathbf{w}_{T_c}^c]$ represent the sequences of word representations obtained from the last layer of the PLM for the question q and the answer passage $c \in C$, respectively.

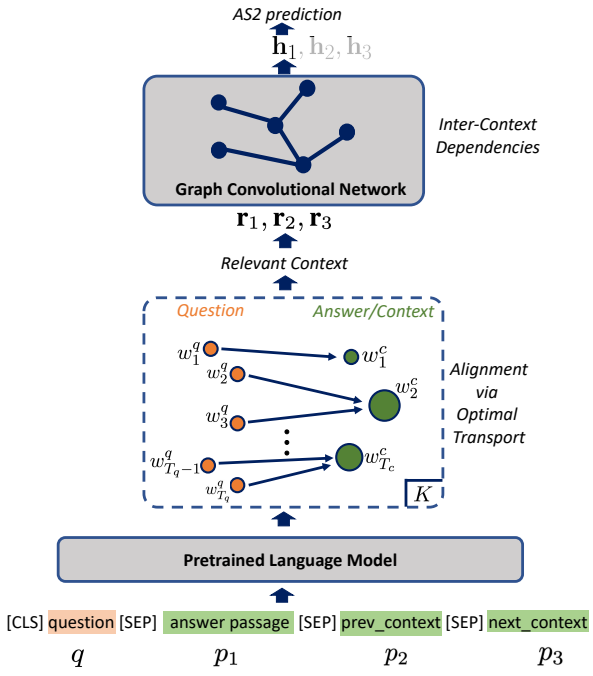

Fig. 2: A diagram depicting the knowledge retriever in our framework for KARP.

Question-Context Alignment with OT In this section, we present our approach for identifying relevant context within the answer passage and its surrounding passages based on the alignment of words with the question. Specifically, we introduce the use of Optimal Transport (OT) [36, 9] to address the task of aligning the question with the context for DPR.

OT is a well-established technique used to transfer probability from one distribution to another by establishing an alignment between two sets of points. In the discrete setting, we are provided with two probability distributions, denoted as p_X and p_Y, defined over two sets of points, namely $X = \{x_i\}_{i=1}^{n}$ and $Y = \{y_j\}_{j=1}^{m}$ ($\sum_i p_{x_i} = 1$ and $\sum_j p_{y_j} = 1$). Additionally, a distance function $D(x, y) : X \times Y \rightarrow \mathbb{R}^+$ is given to quantify the dissimilarity between any two points x and y. The objective of OT is to determine a mapping that transfers the probability mass from the points in $\{x_i\}_{i=1}^{n}$ to the points in $\{y_j\}_{j=1}^{m}$, while minimizing the overall cost associated with this transportation. Formally, this involves finding the transportation matrix $\pi_{XY} \in \mathbb{R}^{+\,n \times m}$ that minimizes the following transportation cost:

$$d_{XY} = \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} D(x_i, y_j)\pi_{XY\,ij}, \tag{1}$$

so that $\pi_{XY}\mathbf{1}_m = p_X$ and $\pi_{XY}^T\mathbf{1}_n = p_Y$. The transportation matrix π_{XY} signifies the best matching between the sets of points X and Y, where each row i in the matrix indicates the optimal alignment from a point $x_i \in X$ to each point $y_j \in Y$.

In our problem of aligning the question with the answer passage, we treat the question q and the answer/context passage c as two point sets: $\{w_i^q\}_{i=1}^{T_q}$ and $\{w_i^c\}_{i=1}^{T_c}$ respectively (each word is a point)[5]. To determine the probability distributions for these word sets, we propose calculating the word frequencies and then normalizing the sum of frequencies. Specifically, the probability distribution for the question is obtained by:

$$p_{w_i^q} = \frac{freq(w_i^q)}{\sum_{i'=1}^{T_q} freq(w_{i'}^q)} \tag{2}$$

The frequency $freq(w_i^q)$ corresponds to the number of occurrences of the word w_i^q in the training data's questions. The same approach is applied to the answer/context passage. To handle unseen words during testing, we utilize Laplace smoothing to assign a non-zero probability. Moving on, we estimate the distance between two words $w_i^q \in q$ and $w_j^c \in c$ by measuring their semantic divergence, which involves computing the Euclidean distance between their contextualized representations obtained from the PLM: $D(w_i^q, w_j^c) = ||\mathbf{w}_i^q - \mathbf{w}_j^c||$. The Sinkhorn-Knopp algorithm is then efficiently employed to solve for the optimal transportation matrix π_{XY} (in this case, π_{qc} for the question q and the passage c) [55, 9]. Finally, we obtain the relevant context r_c for the passage c by taking the union of words w_j^c that have the highest transportation probabilities:

$$r_c = \bigcup_{i=1}^{T_q} \{w_j^c | j = \text{argmax}_{1 \leq j' \leq T_c} \pi_{qc\,ij'}\} \tag{3}$$

[5] Before performing the alignment, we remove stopwords and punctuation marks from both sets of words.

To compute the representation for the passage c, we take the average sum of the word representations within the relevant context:

$$\mathbf{r}_c = \frac{1}{|r_c|} \sum_{j|w_j^c \in r_c} \mathbf{w}_j^c \tag{4}$$

By incorporating the relevant context, our intention is to eliminate any disruptive or unrelated details from the passage representation.

Answer-Context Dependencies For convenience, let $[\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ denote the representations acquired from Equation (4) for the answer passage $p_1 \equiv c$, the previous passage $p_2 \equiv c_{prev}$, and the next passage $p_3 \equiv c_{next}$. To capture the relationships between these passages, we view each passage as a node in a fully-connected graph $G = (V, E)$, where $V = \{p_i\}$ $(1 \leq i \leq 3)$ is the node set and $E = \{(p_i, p_j)\}$ $(1 \leq i, j \leq 3)$ is the edge set. Our objective is to determine a weight $\alpha_{ij} \in (0, 1)$ for each edge (p_i, p_j) that reflects the dependency of p_i on p_j. To accomplish this, we propose to leverage their semantic representations $\mathbf{r}_i$, $\mathbf{r}_j$, and transportation costs to the question d_{qp_i}, d_{qp_j} to measure the dependency weight α_{ij} between the passages p_i and p_j. Specifically, we first compute the score: $u_{ij} = FFN_{DEP}([\mathbf{r}_i \odot \mathbf{r}_j; d_{qp_i}; d_{qp_j}])$, where $\odot$ is the element-wise product, $[;]$ represents the concatenation operation, and FFN_{DEP} is a feed-forward network. Subsequently, the weight α_{ij} for the edge (p_i, p_j) is obtained through a softmax function:

$$\alpha_{ij} = \frac{\exp(u_{ij})}{\sum_{j'=1}^{K} \exp(u_{ij'})} \tag{5}$$

The derived weights $\{\alpha_{ij}\}$ are subsequently utilized to enrich the passage representations through L layers of a Graph Convolutional Network (GCN) [25]:

$$\mathbf{h}_i^l = \text{ReLU}(\sum_{j=1}^{K} \alpha_{ij} \mathbf{W}^l \mathbf{h}_j^{l-1} + \mathbf{b}^l) \tag{6}$$

where $\mathbf{W}^l$, $\mathbf{b}^l$ are learnable weight matrix and bias for the layer l of the GCN $(1 \leq l \leq L)$, and $\mathbf{h}_i^0 \equiv \mathbf{r}_i$ is the input representation for the passage p_i. The output vectors $\mathbf{h}_i^L \equiv \mathbf{h}_i$ at the last layer of the GCN serve as the final representations for the passages p_i. Intuitively, the weights α_{ij} enable each passage to decide the amount of information it receives from the other passages to improve its representation for the task. The representation $\mathbf{h}_1$ for the answer passage $p_1 \equiv c$ is finally sent to a feed-forward network with a sigmoid output function to estimate the correctness score $p_c \in (0, 1)$ for the answer passage c: $p_c = FFN_{DPR}(\mathbf{h}_1)$. For training, we minimize the binary cross-entropy loss with the correctness scores p_c. At inference time, consistent with previous research [13], we include all answer passages for each question for ranking.

2.3 Generative LLM-based Question Answering Model

Background on Text Generation Finetuning Text generation finetuning has become a general approach to solving different NLP tasks, where input and expected output of a

task can be represented as source and target text respectively for a generative model to learn the task [44, 34, 29]. For example, a pretrained generative LLM such as BART [28] and T5 [44] can be finetuned on sentiment analysis by taking a text statement (e.g., *"I really like the story"*) as source text to generate a text label (i.e., *"Positive"*, *"Negative"*, *"Neutral"*) to indicate the sentiment of the statement. As the text generation resembles the pretraining tasks (e.g., predicting next words) for the generative LLMs, the formulation could facilitate the transfer learning to the target task. In addition, it enables the data augmentation method where training data for a task may also be leveraged for another task if the two tasks both are convertable to the text generation format [31]. These advantages have led to significant performance improvements for many NLP tasks such as event extraction [31], named entity recognition [66], and dependency parsing [29]. Similar to other NLP tasks, the generative methods have been explored for improving QA performance [21, 49, 43, 19, 27, 18, 12, 37]. To avoid hallucination and improve factual accuracy for the models, recent works on generative QA employ the retrieval-based methods such as GenQA [17] and WebGPT [38].

GenQA is introduced by Hsu et al. [17] for generating appropriate answers for user questions instead of simply choosing the best answer candidate. This expands the answer retrieval pipeline with an additional generation stage to produce correct and satisfactory answers, even in cases where a highly ranked candidate is not acceptable or does not provide a natural response to the question. In particular, GenQA employs a pretrained generative LLM to produce an answer by taking a given question and a list of answer candidates as input, sorted by a trained reranker system.

WebGPT is designed by OpenAI researchers [38] to tackle the problem of long-form question-answering, which involves generating a paragraph-length answer to an open-ended question. Specifically, WebGPT uses the Microsoft Bing Web Search API to retrieve relevant documents for a given question. The model then interacts with a text-based environment where it can take actions such as clicking on links or opening new web pages to locate relevant passages from which to generate answers.

Our Proposed Finetuning Strategy The main goal of a general text-generation model is to produce an output text sequence $\mathbf{y} = [y_1, y_2, \ldots, y_T]$ based on a given input text sequence $\mathbf{x} = [x_1, x_2, \ldots, x_S]$, where the lengths of the input and output sequences are denoted by S and T, respectively. With a pretrained encoder-decoder LLM such as BART [28] or T5 [44], we can compute the conditional probability of $P(\mathbf{y}|\mathbf{x})$ for training the model. At test time, the decoder merges the previous output and input text to create the current output. A decoding algorithm such as Greedy or Beam Search [63] can be used to generate an output text with the highest likelihood. For QA, given a question q and a retrieved web content c (e.g., top answer passages), previous works such as GenQA and WebGPT are trained to take (q, c) for as the source sequence to produce a response as the target sequence $a = f_\theta(q, c)$, where f_θ denotes the corresponding LLM with the parameters θ. As a result, f_θ may merely learn to copy/synthesize information from c to produce a if c often contains necessary information for correctly answering the question q in training data. Relying solely on the retrieved content c, the model

may fail to provide a correct answer for a given question if c is missing or contains irrelevant/noisy information. In other words, performance of these retrieval-based QA models are limited to an upper bound by the knowledge retriever.

Different from the previous works that follow a single-style "$a = f_\theta(q, c)$" finetuning strategy, we propose to employ a multi-style finetuning strategy, where both "$a = f_\theta(q, c)$" and "$a = f_\theta(q)$" are used to train the model. The latter intentionally excludes the external knowledge c from the input to encourage the model to retrieve its own knowledge from the model parameters θ, which have been pretrained on massive unlabeled text data [28, 44, 11, 56]. To combine the two finetuning styles, we propose to finetune the LLM with "$a = f_\theta(q, c)$", and sequentially finetune the model with "$a = f_\theta(q)$". In this way, our model does not completely rely on the retrieval results to generate answers for given questions. At test time, we use the "$a = f_\theta(q, c)$" style to make predictions. The retrieved content c now can be considered as a source of external knowledge along with the pretrained knowledge contained in the model parameters θ to generate an answer for the question. Under this perspective, we consider various QA datasets for each step in our finetuning process. We call such dataset collection OKQA as they are publicly available and contains high-quality knowledge.

MS Marco QA NLG is a specialized version of the MS Marco dataset [39] that aims to produce natural language responses to user inquiries using web search result excerpts. This dataset includes $182K$ queries from Bing search logs, each is associated with top ten most relevant passages. A human annotator is then required to look at the passages and synthesize an answer using the content of the passages that most accurately addresses the query.

Super Natural Instructions (SNI) is a data collection proposed by [60]. The corpus consists of $1,616$ diverse NLP tasks and their expert-written instructions. In this work, we consider only question-answering tasks such as extractive QA with SQUAD [45] and multiple-choice QA with MCTest [48]. For each task, we consider anything but a question q provided in the input as context c. Particularly, the context c can be a passage, a fact, or a set of answer choices associated with the question. As a result, we obtain $180K$ examples for finetuning our model.

Anthropic is introduced by [1], containing conversations between a human and a computer assistant. For each conversation, we consider a human question and the previous question (if any) as the input sequence and the answer from the assistant as the output sequence. As questions in a conversation are usually related to each other, the previous question can be considered as a form of relevant context c for clarifying the current question q. Consequently, we obtain $280K$ examples for finetuning our model.

Dense Passage Retrieval datasets, namely, WikiQA [69] and WDRASS [73] are also used for finetuning our model. WikiQA is a collection of questions and answer candidates that have been manually annotated using Bing query logs on Wikipedia. WDRASS is a large-scale dataset of questions that are non-factoid in nature, such as questions that begin with "why" or "how". The dataset contains around $64,000$ questions and over $800,000$ labeled passages that have been extracted from a total of $30M$ documents. Each

question in such DPR datasets is associated with a set of answer candidates, in which some of the candidates are correct answers. As a question can have multiple correct answers, we select the longest answer as the output sequence for the question, which is considered as the input sequence. This results in a set of $105K$ examples for finetuning our model.

In the end, the datasets where context is available for a question are employed in the step 1 of our finetuning process while the other datasets are used for further training the model in the subsequent step. With a huge amount of various QA tasks, we expect this could teach the model to understand the nature of question answering and how to utilize its own parametric knowledge (in case no context is provided) and external knowledge (i.e., relevant context) to answer a given question.

3 Experiments

3.1 Benchmarking the Knowledge Retriever

Experimental Setup

Datasets We follow the previous work [13, 74] to conduct the evaluation. In particular, we use (i) **WikiQA** [69], consisting of questions from Bing query logs and manually annotated answers from Wikipedia, and (ii) **WDRASS** [74], a large-scale web-based dataset having factoid and non-factoid questions, to investigate our retrieval performance. We use the same train/dev/test splits used in previous work.

Hyper-parameters and Tools In accordance with previous work, we use a small portion of the WikiQA training data to tune hyper-parameters for our model and select the best hyper-parameters for all the datasets [26]. We employ Adam optimizer to train the model with a learning rate of $1e$-5 and a batch size of 64. We set 400 for the hidden vector sizes for all the feed-forward networks, $L = 2$ for the number of the GCN layers. We use Pytorch version 1.7.1 and Huggingface Transformers version 3.5.1 To implement the models. We use the NLTK library version 3.5 [4] to preprocess the data and remove stopwords. The model performance is obtained over three runs with random seeds.

Evaluation Metrics We measure the model performance using the following standard metrics: Precision-at-1 (P@1) and Mean Average Precision (MAP) on the entire set of answer candidates for each question.

Performance Comparison We compare our proposed model with TANDA [13], which is the current state-of-the-art model. Table 2 shows the performance comparison between the models on two settings: i) using a non-finetuned RoBERTa-Base encoder, and ii) using a fine-tuned RoBERTa-Base encoder. The non-finetuned RoBERTa-Base is obtained from [33] while the other is produced by fine-tuning TANDA on the ASNQ dataset [13]. As can be seen from the table, all the models benefit from using the finetuned RoBERTa-Base encoder. Across the two settings, our model outperforms the previous models by large margins, demonstrating its effectiveness for the task.

Model	WikiQA				WDRASS	
	w/o ASNQ		with ASNQ		with ASNQ	
	P@1	MAP	P@1	MAP	P@1	MAP
TANDA	63.24*	75.00*	78.67*	86.74*	54.60	63.50
Ours	**74.16**	**83.29**	**83.77**	**89.28**	**55.9**	61.8

Table 2: Performance comparison on WikiQA and WDRASS, * indicates results reported by [26].

In Table 2, we show the performance of our proposed model compared to TANDA on the WDRASS test set. As we can see, our knowledge retriever significantly improves the performance for P@1 score, however, decreases the performance for MAP score. We attribute this to the fact that questions in WDRASS dataset usually have more than 1 correct answers for a single question while our model ranks the answer candidates individually. However, we note that the top-1 answer candidate is often the most helpful for the answering process.

3.2 Evaluation for Knowledge-Aware Answer Generation

Experimental Setup

Dataset We acquire the evaluation data as follows. First, we randomly select 2,000 questions from the MS MARCO QA NLG test set. For each question, we rank all the context passages using our model trained on WDRASS to obtain the top 5 candidates. We then concatenate the question and candidates to form the input, which is used to generate the predicted answer.

Evaluation Metrics We employ widely-used evaluation metrics, including ROUGE [30], BLEU [40], and BERTScore [72], for assessing the quality of generated answers in comparison to human-written natural answers. These metrics are commonly applied to standard text generation tasks such as summarization [71], machine translation [57], and answer generation [43].

It is important to note that these metrics have their own limitations; however, these can be mitigated by providing more and higher-quality reference texts [5]. In the context of answer generation, we enhance the reliability of these measurements by employing human-written answers as references. Specifically, annotators create the reference answers used in this benchmark after being provided with the candidate responses.

Performance Comparison Table 3 presents a comparison of three different configurations of KARP with GenQA model in terms of BLEU, RougeL, and BERTScore metrics.

The results demonstrate that all three KARP configurations outperform the GenQA model across all evaluation metrics. The best-performing configuration (config 2) achieves a BLEU score of 39.4, a RougeL score of 0.608, and a BERTScore of 0.752. These results indicate that KARP offers a significant improvement over the GenQA

Model	BLEU	RougeL	BERTScore
GenQA [17]	14.6	0.518	0.698
KARP (config 1)	38.3	0.632	0.762
KARP (config 2)	39.4	0.608	0.752
KARP (config 3)	38.9	0.604	0.750

Table 3: Comparison of our three KARP models trained with different hyper-parameter settings to GenQA [17].

model in the context of answer generation, which we attribute to our specialized fine-tuning strategy for QA.

3.3 End-to-end Evaluation for Knowledge-Aware Response Planning

In this section, we evaluate KARP in an end-to-end industry-scale scenario.

Experimental Setup We outline the experimental setup to evaluate the end-to-end performance of KARP in a web-scale scenario, involving tens of millions of web documents. The configuration allows us to study the scalability and effectiveness of our approach in a real-world, large-scale setting.

Web Document Collection We constructed a large collection of web data, comprising documents and passages, to facilitate the development of knowledge retrieval for end-to-end system evaluation. This resource enables us to assess the impact of our work in an industry-scale ODQA setting. We selected English web documents from the top 5,000 domains, including Wikipedia, from Common Crawl's 2019 and 2020 releases. The pages were split into passages following the DPR procedure [23], limiting passage length to 200 tokens while maintaining sentence boundaries. This produced a collection of roughly 100 million documents and 130 million passages. From this, we built (i) a standard Lucene/Elasticsearch index and (ii) a neural-based DPR index [23].

Web-scale Knowledge Retrieval For each question, we retrieved up to 1,000 documents/passages using both indexes. We then rank the passages and applied our knowledge retriever to select relevant passages. We used top $K = 5$ candidates as external knowledge for a question.

Question Sampling We randomly selected 2,000 questions from WDRASS test set as it shows to represent natural questions extracted from the Web. In addition, the questions were also manually labeled.

Baselines We employ GenQA [17] as our main baseline in this experiment. We compare the performance of our system obtained by our proposed fine-tuning strategy and the standard fine-tuning (i.e., combining all datasets for finetuning) in a data parity setting.

Evaluation Metrics We evaluate the performance of the end-to-end QA system using accuracy metrics, i.e., the percentage of questions that were answered satisfactorily. Additionally, we define a correct answer as one that must not only be factually accurate, but also expressed in a natural and fluent manner. Answers that are too verbose or oddly phrased are considered unsatisfactory.

Performance Comparison The result show in the following table Table 4.

Model	Accuracy
TANDA [13]	*baseline*
GenQA [17]	+2.20%
KARP → MS MARCO	+6.20%
KARP → OKQA	+7.40%

Table 4: Relative accuracy of different QA settings: TANDA [13], GenQA [17], and our proposed frame work for KARP in two data configurations: MS MARCO (data parity) and OKQA.

Table 4 presents the relative accuracy of different QA settings, including TANDA [13], GenQA [17], and our proposed KARP with two data configurations: MS MARCO (data parity) and (robust fine-tuning). From the table, we observe that GenQA outperforms TANDA by 2.20%. Our proposed KARP model achieves even better results, with a 6.20% increase in accuracy when using the MS MARCO data configuration and a 7.40% increase in accuracy when using OKQAconfiguration. This demonstrates the effectiveness of our proposed KARP model in various data settings.

4 Related Work

Large Language Models (LLMs) LLMs have transformed NLP technologies with the advent of the Transformer architecture [57]. Two fundamental pre-training objectives, Masked Language Modeling (MLM) and Causal Language Modeling (CLM), underpin the success of these models. MLM, introduced by BERT [10], predicts masked tokens in a sentence using surrounding context, enabling LLMs to learn bidirectional representations that excel in various NLP tasks. In contrast, CLM, exemplified by GPT [41], predicts the next token in a sequence given its preceding context, showing remarkable success in text generation and other downstream applications [42, 22, 43]. In this paper, we leverage the CLM architecture for its language generation capabilities to enhance QA performance.

General Question Answering using LLM A standard QA system consists of (i) a retrieval engine that returns relevant knowledge and (ii) a model that generates a response addressing the question, either through selection [52, 70, 13] or abstractive summarization of the top-selected answers [18, 12, 37]. In particular, recent summarization-based approaches, e.g., GenQA [18, 12, 37], are highly susceptible to hallucination due to the

absence of special treatment of irrelevant candidates, which commonly appear among the top-ranked options. As a result, the generated answer may seem plausible but could be factually incorrect [20, 76, 75, 64, 58, 54, 47, 46]. Even though its original goal is to generate more natural answers, GenQA [18, 12, 37] can be considered as a method to ground LLMs for QA as it decodes an answer from the concatenation of both question and answer candidates. This approach, however, requires good answer candidates and careful finetuning to reduce hallucinations.

We propose, instead, a novel generation-based approach that leverages the emerging language reasoning capabilities of Large Language Models (LLMs) [41] to enhance quality of generated answers. In particular, KARP is designed to mitigate the reliance on oracle data by making use of the context, such as all choices in multiple-choice QA, instead of a correct answer alone, i.e., the correct choice. The experiments demonstrated that our proposed framework for KARP is highly resilient to noisy input data, and bring about broader application across different QA tasks.

Fine-tuning Strategies for LLMs Several fine-tuning strategies have been specifically proposed for large language models (LLMs). These strategies can be broadly categorized into two groups: architecture-centric and data-centric. (i) Architecture-centric fine-tuning aims to improve the model's robustness and adaptability by modifying hyper-parameters across layers. Gradual unfreezing [16] is one example, involving sequential fine-tuning of model layers to prevent catastrophic forgetting and better adapt to downstream tasks. Layer-wise learning rate decay [41] is another example, where different learning rates are assigned to various layers to enable more refined adaptation to the target task. (ii) Data-centric fine-tuning, on the other hand, concentrates on leveraging data from different sources or intermediate tasks to enhance model performance. Sequential fine-tuning [14, 13] involves training the model on intermediate tasks before the final target task, improving its performance on the latter. Combining several related datasets for multi-task fine-tuning has also been shown to improve performance on the target task [32]. Our work is related to data-centric fine-tuning. In particular, we propose a novel strategy specifically designed for the question answering context. By leveraging both external knowledge and intrinsic parametric knowledge of LLMs, our approach aims to enhance the quality of generated answers in QA tasks.

5 Conclusion

In this paper, we presented a novel framework powered by large language models (LLMs) for KARP. To that end, we proposed an efficient fine-tuning strategy for KARP that leverages (i) the emergent language reasoning abilities of LLMs and (ii) general question answering advances, including modelings and resources. Our experimental results show that KARP improves the state of the art in general QA tasks and outperforms vanilla fine-tuning of LLMs in a dataset-parity setting. This research highlights the significance of leveraging the intrinsic parametric knowledge of LLMs rather than relying solely on conventional sequence-to-sequence fine-tuning, in order to improve their performance in question answering tasks.

References

1. Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al.: Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862 (2022)
2. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers. pp. 2503–2514 (2016)
3. Bao, J., Duan, N., Zhou, M., Zhao, T.: Knowledge-based question answering as machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 967–976 (2014)
4. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
5. Callison-Burch, C., Osborne, M., Koehn, P.: Re-evaluating the role of Bleu in machine translation research. In: 11th Conference of the European Chapter of the Association for Computational Linguistics. pp. 249–256. Association for Computational Linguistics, Trento, Italy (Apr 2006)
6. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051 (2017)
7. Chen, D., Yih, W.t.: Open-domain question answering. In: Proceedings of the 58th annual meeting of the association for computational linguistics: tutorial abstracts. pp. 34–37 (2020)
8. Clark, P., Etzioni, O., Khot, T., Sabharwal, A., Tafjord, O., Turney, P., Khashabi, D.: Combining retrieval, statistics, and inference to answer elementary science questions. Proceedings of the AAAI Conference on Artificial Intelligence **30**(1) (Mar 2016). https://doi.org/10.1609/aaai.v30i1.10325
9. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
11. FitzGerald, J.G.M., Ananthakrishnan, S., Arkoudas, K., Bernardi, D., Bhagia, A., Bovi, C.D., Cao, J., CHADA, R., Chauhan, A., Chen, L., Dwarakanath, A., Dwivedi, S., Gojayev, T., Gopalakrishnan, K., Gueudre, T., Hakkani-Tür, D., Hamza, W., Hueser, J., Jose, K.M., Khan, H., Liu, B., Lu, J., Manzotti, A., Natarajan, P., Owczarzak, K., Oz, G., Palumbo, E., Peris, C., Prakash, C.S., Rawls, S., Rosenbaum, A., Shenoy, A., Soltan, S., Harakere, M., Tan, L., Triefenbach, F., Wei, P., Yu, H., Zheng, S., Tur, G., Natarajan, P.: Alexa teacher model: Pretraining and distilling multi-billion-parameter encoders for natural language understanding systems. In: KDD 2022 (2022)
12. Gabburo, M., Koncel-Kedziorski, R., Garg, S., Soldaini, L., Moschitti, A.: Knowledge transfer from answer ranking to answer generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 9481–9495. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022)
13. Garg, S., Vu, T., Moschitti, A.: Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. Proceedings of the AAAI Conference on Artificial Intelligence **34**(05), 7780–7788 (Apr 2020). https://doi.org/10.1609/aaai.v34i05.6282
14. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't stop pretraining: Adapt language models to domains and tasks. In: Proceedings of the

58th Annual Meeting of the Association for Computational Linguistics. pp. 8342–8360. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.740
15. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. Advances in neural information processing systems **28** (2015)
16. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 328–339. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1031
17. Hsu, C.C., Lind, E., Soldaini, L., Moschitti, A.: Answer generation for retrieval-based question answering systems. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4276–4282. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.374
18. Hsu, C.C., Lind, E., Soldaini, L., Moschitti, A.: Answer generation for retrieval-based question answering systems. In: ACL Findings 2021 (2021)
19. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 874–880. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.74
20. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12) (mar 2023). https://doi.org/10.1145/3571730
21. Jiang, Z., Araki, J., Ding, H., Neubig, G.: Understanding and improving zero-shot multi-hop reasoning in generative question answering. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 1765–1775. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (Oct 2022)
22. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)
23. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550
24. Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., Hajishirzi, H.: UNIFIEDQA: Crossing format boundaries with a single QA system. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1896–1907. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.171
25. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations (2017)
26. Lauriola, I., Moschitti, A.: Answer sentence selection using local and global context in transformer models. In: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I. pp. 298–312. Springer (2021)
27. Lewis, M., Fan, A.: Generative question answering: Learning to answer the whole question. In: International Conference on Learning Representations (2019)
28. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: 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. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703
29. Lin, B., Yao, Z., Shi, J., Cao, S., Tang, B., Li, S., Luo, Y., Li, J., Hou, L.: Dependency parsing via sequence generation. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 7339–7353. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022)
30. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
31. Liu, J., Chen, Y., Liu, K., Bi, W., Liu, X.: Event extraction as machine reading comprehension. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1641–1651. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.128
32. Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4487–4496. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1441
33. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
34. Lu, Y., Lin, H., Xu, J., Han, X., Tang, J., Li, A., Sun, L., Liao, M., Chen, S.: Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 2795–2806. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.217
35. Maynez, J., Narayan, S., Bohnet, B., McDonald, R.: On faithfulness and factuality in abstractive summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1906–1919. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.173
36. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Mem. Math. Phys. Acad. Royale Sci. pp. 666–704 (1781)
37. Muller, B., Soldaini, L., Koncel-Kedziorski, R., Lind, E., Moschitti, A.: Cross-lingual open-domain question answering with answer sentence generation. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 337–353. Association for Computational Linguistics, Online only (Nov 2022)
38. Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al.: Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332 (2021)
39. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human generated machine reading comprehension dataset (November 2016)
40. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002). https://doi.org/10.3115/1073083.1073135
41. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
42. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

43. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1) (jan 2020)
44. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020)
45. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2383–2392. Association for Computational Linguistics, Austin, Texas (Nov 2016). https://doi.org/10.18653/v1/D16-1264
46. Raunak, V., Menezes, A., Junczys-Dowmunt, M.: The curious case of hallucinations in neural machine translation. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1172–1183. Association for Computational Linguistics, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.92
47. Rebuffel, C., Roberti, M., Soulier, L., Scoutheeten, G., Cancelliere, R., Gallinari, P.: Controlling hallucinations at word level in data-to-text generation. CoRR **abs/2102.02810** (2021)
48. Richardson, M., Burges, C.J., Renshaw, E.: MCTest: A challenge dataset for the open-domain machine comprehension of text. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 193–203. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013)
49. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5418–5426. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.437
50. Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E.M., Boureau, Y.L., Weston, J.: Recipes for building an open-domain chatbot. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 300–325. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.24
51. Saxena, A., Chakrabarti, S., Talukdar, P.: Question answering over temporal knowledge graphs. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 6663–6676. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.520
52. Severyn, A., Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp. 373–382 (2015)
53. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 3784–3803. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.320
54. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 3784–3803. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.320
55. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967)
56. Soltan, S., Ananthakrishnan, S., FitzGerald, J.G.M., Gupta, R., Hamza, W., Khan, H., Peris, C., Rawls, S., Rosenbaum, A., Rumshisky, A., Prakash, C.S., Sridhar, M., Triefenbach, F., Verma,

A., Tur, G., Natarajan, P.: Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. arXiv (2022)
57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
58. Wang, C., Sennrich, R.: On exposure bias, hallucination and domain shift in neural machine translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 3544–3552. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.326
59. Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M.: Gated self-matching networks for reading comprehension and question answering. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 189–198 (2017)
60. Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Naik, A., Ashok, A., Dhanasekaran, A.S., Arunkumar, A., Stap, D., Pathak, E., Karamanolakis, G., Lai, H., Purohit, I., Mondal, I., Anderson, J., Kuznia, K., Doshi, K., Pal, K.K., Patel, M., Moradshahi, M., Parmar, M., Purohit, M., Varshney, N., Kaza, P.R., Verma, P., Puri, R.S., Karia, R., Doshi, S., Sampat, S.K., Mishra, S., Reddy A, S., Patro, S., Dixit, T., Shen, X.: Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 5085–5109. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022)
61. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., Fedus, W.: Emergent abilities of large language models. Transactions on Machine Learning Research (2022), survey Certification
62. Weston, J., Bordes, A., Chopra, S., Rush, A.M., Van Merriënboer, B., Joulin, A., Mikolov, T.: Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698 (2015)
63. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1296–1306. Association for Computational Linguistics, Austin, Texas (Nov 2016). https://doi.org/10.18653/v1/D16-1137
64. Xiao, Y., Wang, W.Y.: On hallucination and predictive uncertainty in conditional language generation. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 2734–2744. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.236
65. Xu, J., Wang, Y., Tang, D., Duan, N., Yang, P., Zeng, Q., Zhou, M., Sun, X.: Asking clarification questions in knowledge-based question answering. 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). pp. 1618–1629 (2019)
66. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various NER subtasks. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 5808–5822. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.451
67. Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., Lin, J.: End-to-end open-domain question answering with BERTserini. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 72–77. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-4013

68. Yang, Y., Yih, W.t., Meek, C.: Wikiqa: A challenge dataset for open-domain question answering. In: Proceedings of the 2015 conference on empirical methods in natural language processing. pp. 2013–2018 (2015)
69. Yang, Y., Yih, W.t., Meek, C.: WikiQA: A challenge dataset for open-domain question answering. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2013–2018. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015)
70. Yoon, S., Dernoncourt, F., Kim, D.S., Bui, T., Jung, K.: A compare-aggregate model with latent clustering for answer selection. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2093–2096 (2019)
71. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 11328–11339. PMLR (13–18 Jul 2020)
72. Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. In: International Conference on Learning Representations (2020)
73. Zhang, Z., Vu, T., Gandhi, S., Chadha, A., Moschitti, A.: Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4707–4711 (2022)
74. Zhang, Z., Vu, T., Gandhi, S., Chadha, A., Moschitti, A.: Wdrass: A web-scale dataset for document retrieval and answer sentence selection. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. p. 4707–4711. CIKM '22, Association for Computing Machinery (2022)
75. Zhao, Z., Cohen, S.B., Webber, B.: Reducing quantity hallucinations in abstractive summarization. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2237–2249. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.203
76. Zhou, C., Neubig, G., Gu, J., Diab, M., Guzmán, F., Zettlemoyer, L., Ghazvininejad, M.: Detecting hallucinated content in conditional neural sequence generation. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 1393–1404. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.120