In-context Learning with Retrieved Demonstrations for Language Models: A Survey

Anonymous authors
Paper under double-blind review

Abstract

Language models, especially pre-trained large language models, have showcased remarkable abilities as few-shot in-context learners (ICL), adept at adapting to new tasks with just a few demonstrations in the input context. However, the model's ability to perform ICL is sensitive to the choice of the few-shot demonstrations. Instead of using a fixed set of demonstrations, one recent development is to *retrieve* demonstrations tailored to each input query. The implementation of demonstration retrieval is relatively straightforward, leveraging existing databases and retrieval systems. This not only improves the efficiency and scalability of the learning process but also has been shown to reduce biases inherent in manual example selection. In light of the encouraging results and growing research in ICL with retrieved demonstrations, we conduct an extensive review of studies in this area. In this survey, we discuss and compare different design choices for retrieval models, retrieval training procedures, and inference algorithms.

1 Introduction

Few-shot in-context learning (ICL) is the ability of large language models (LLMs) to perform a new task when a few input-output examples, or *demonstrations*, for the new task are given alongside the actual task input. Importantly, the model parameters do not have to be fine-tuned towards the new task. ICL is popularized by the work on pre-trained large language models, which can perform ICL without being trained to do so (Brown et al., 2020), though smaller language models can also be explicitly trained to perform ICL (Min et al., 2022a).

ICL presents several advantages over the conventional methodology for adapting language models to a downstream task, which typically involves initial pre-training followed by subsequent fine-tuning. One significant merit of ICL is the circumvention of fine-tuning, which might not always be possible due to limited access to the model parameters or constraints on computational resources (Brown et al., 2020). Furthermore, ICL avoids common issues associated with fine-tuning, such as overfitting(Ying, 2019; Kazemi et al., 2023a). Compared to parameter-efficient fine-tuning methods (PEFT) (Hu et al., 2021; Dettmers et al., 2023; Lester et al., 2021), ICL is computationally cheaper and remain the model parameters unchanged thus preserving the generality of the LLMs.

Early ICL implementations use a fixed set of demonstrations for each target task. These demonstrations could be hand-crafted by human (Hendrycks et al., 2021; Wei et al., 2022; Kazemi et al., 2023b), randomly chosen from training data (Brown et al., 2020; Lewkowycz et al., 2022). Beyond random selection, there are more advanced selection processes based on metrics such as complexity (Fu et al., 2022), diversity (Li & Qiu, 2023a), difficulty (Drozdov et al., 2023), concept learning (Wang et al., 2023b) and perplexity (Gonen et al., 2023). Importantly, the demonstrations remain context-insensitive (i.e. the same demonstrations are used regardless of the query) which could hinder unlocking the true potential of the LLMs. The effectiveness of such demonstrations is influenced by factors such as the quality, quantity, and ordering of the demonstrations.

Retrieval-based ICL (RetICL) presents a paradigm shift in the optimization of language model performance, moving beyond static, pre-defined demonstration sets to a dynamic, context-sensitive approach. At the heart of this innovation is the concept of *adaptive demonstration selection*, where a specialized retriever intelligently

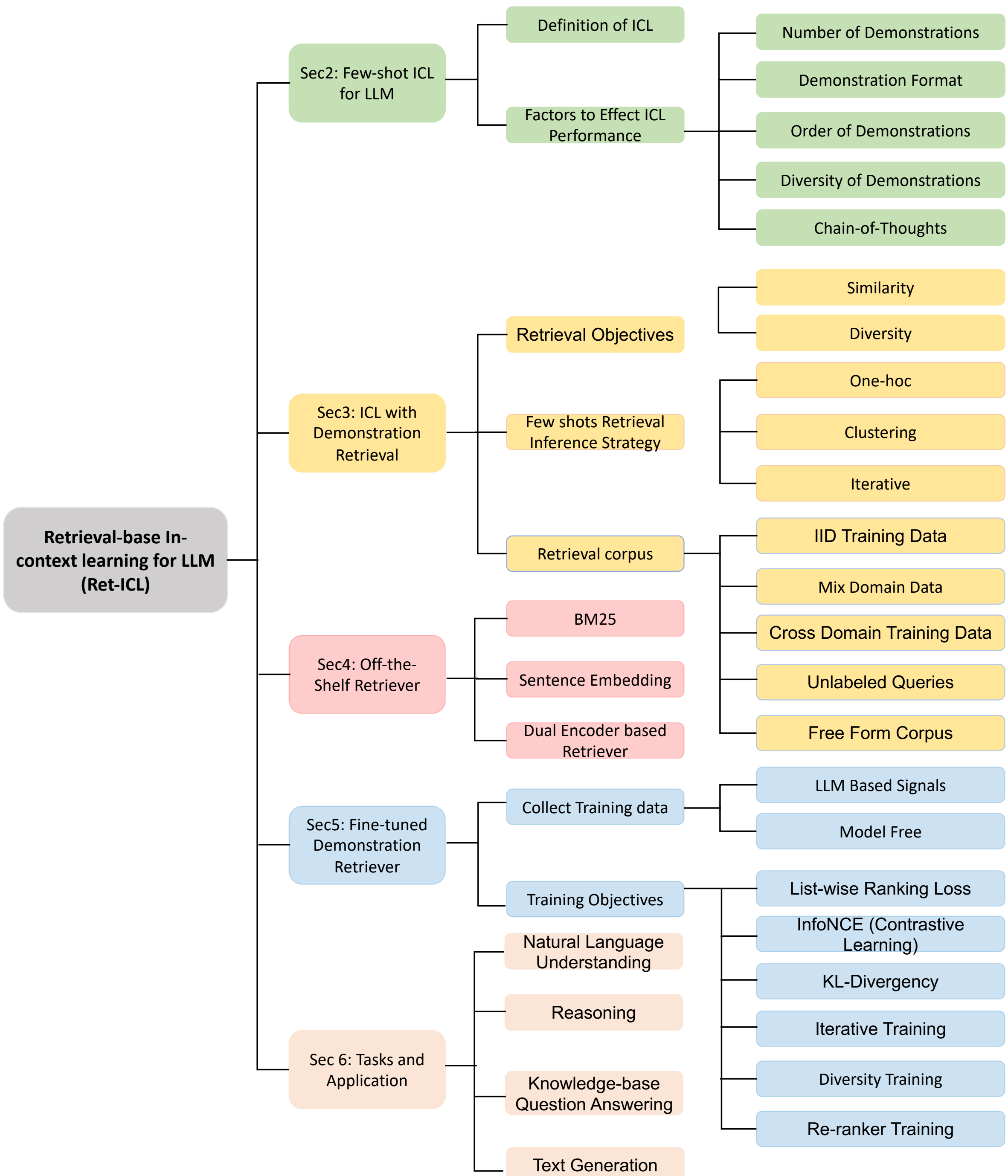

Figure 1: Structure of the Survey.

curates tailored demonstrations for each specific task input. This method has not only consistently outshined approaches relying on random or static hand-crafted demonstrations but has also demonstrated a remarkable resilience to a variety of influencing factors.

The efficacy of RetICL pivots on the "relevance" and "usefulness" of the demonstrations it selects, a process intricately influenced by multiple elements. These include the nature of the retriever—ranging from general off-the-shelf models to finely-tuned, domain-specific variants—the source and diversity of the retrieval corpus, the retriever's objectives (focusing on either similarity or diversity), and the strategies for integrating multiple demonstrations. Over the past two years, numerous and sometimes concurrent works have studied RetICL each with different terminology and with variations in problem definition and subsequent methodologies, making it difficult to comprehend the current state of research and practice in RetICL, especially for newcomers to the field. In this comprehensive survey, we meticulously analyze 22 seminal papers in the field of RetICL, as detailed in Table 1, and provide a categorization of their main building blocks (See Figure 1). Our work not only provides a thorough synthesis of existing research but also underscores the areas where RetICL significantly surpasses previous ICL methods, and illuminates many paths forward for future innovations in this area, thus serving as a critical resource for ICL.

2 Few-shot In-context Learning for Language Models

Language models (LMs) (Zhao et al., 2023; Rosenfeld, 2000) are probabilistic models that assign probabilities to sequences of words and are essential components in many tasks. Let s represent a sequence of words (e.g., a sentence) and $w_1, w_2, \ldots, w_n$ represent the tokens in the sequence. Based on the chain rule, the probability $p(s)$ can be decomposed into the following product of probabilities:

$$p(s) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1}) = \prod_{k=1}^{n} p(w_k \mid w_1, \ldots, w_{k-1})$$

where each element in the product corresponds to the probability of a token given the previous tokens. Based on the above decomposition, an LM can be constructed by learning the probability of the next token given the previous ones.

Earlier LMs were mostly based on N-gram models which are based on the Markovian assumption that the next token only depends on the recent context (Jurafsky, 2021). Based on this assumption, $p(w_k \mid w_1, \ldots, w_{k-1})$ is approximated, e.g., by $p(w_k \mid w_{k-2}, w_{k-1})$ in the case of a bi-gram model; $p(w_k \mid w_{k-2}, w_{k-1})$ is then approximated statistically based on the number of times w_k appeared after w_{k-2} in a large corpora of text, w_{k-1} divided by the total number of times w_{k-2}, w_{k-1} appeared in the corpora.

With the advent of word embeddings (Bengio et al., 2000; Mikolov et al., 2013), neural approaches to language modeling gained more popularity, in which a neural network is used to predict the next token probability. The use of powerful neural networks such as long-short term memory (LSTM) models (Hochreiter & Schmidhuber, 1997) and Transformer models (Vaswani et al., 2017) allowed for predicting the next token probability based on a much longer and a variable length context, thus enabling better estimation of $p(w_k \mid w_1, \ldots, w_{k-1})$.

The increased power of neural LMs led to a new learning paradigm for NLP problems. Historically, the dominant learning paradigm for NLP problems was to train models on task-specific data from scratch. Consequently, for each new task, the model had to learn everything from scratch. This often resulted in poor generalization, especially in the cases where previously unobserved vocabulary was observed at the test time. In the subsequent paradigm, an LM was first pre-trained on a large corpora of text making it learn about how language works and gain a vast amount of knowledge about the world (Petroni et al., 2019; Lin et al., 2020; Sung et al., 2021; Yuan et al., 2023); the pre-trained LM (PLM) was then further finetuned on data from the new tasks (Sarzynska-Wawer et al., 2021; Devlin et al., 2018) thus teaching the general PLM the specifics of the new task. This paradigm often resulted in faster learning and higher predictive performance. It was later shown that further finetuning a PLM on multiple tasks leads to better transfer of knowledge across tasks and may lead to better performance on new tasks (Raffel et al., 2020).

Paper	LLMs	Retrieval Method	Retrieval Corpus	Evaluation Tasks	Retriever Training	Retrieval Strategy
SYNCHRO MESH 2021	GPT-3	SBERT + Target Similarity	In Domain	CodeGen	Target Similarity Tuning	One-hoc
KATE 2022	GPT-3	RoBERTa+kNN	In Domain	SA, Table2Text, QA	Sentence Similarity Embedding	One-hoc
EPR 2022	GPT-J, CODEX, GTP-3	SBERT, BM25, Fine-tuned Retriever	In Domain	SP	InfoNCE Loss(contrastive learning)	One-hoc
Z-ICL 2022	GPT-J, GPT-NeoX, GPT-3	SimCSE (Sentence Embeddings)	Free Form Corpus	SA	Sentence Embeddings Similarity	One-hoc
Mem Prompt 2022	GPT-3	Query Transformation + SBERT	In Domain with Human Feedback	Word Reasoning, Ethical Reasoning	Sentence Similarity Embeddings	One-hoc
Teach Me 2022	T5-11B	BM25	In Domain with Human Feedback	QA	Term-based Similarity	One-hoc
IC-DST 2022	GPT-3	Fine-tuned SBERT Retriever	In Domain	DST-to-SQL	Contrastive Learning	One-hoc
Vote-k 2022	GPT-Neo, GPT-J,GPT-3, CODEX	SBERT	Selected subsets of In Domain, labeled	SA, NLI, QA, Summ, CSR, RC	Sentence Embeddings Similarity	One-hoc
Auto-CoT 2022b	GPT-3	Clustering with SBERT	LLM Generated CoT for In Domain	MathR, QA	Sentence Embeddings Similarity	Clustering
XRICL 2022	CODEX	Fine-tuned Retriever Combined with a Reranker	Cross Domain	Text-to-SQL	Distillation by KL Divergence	One-hoc
DSP 2022	GPT-3	ColBERTv2	In Domain	QA	Token Embeddings Similarity	Iterative
PARC 2022	mBERT	Multilingual SBERT	Cross Domain	SA, TopicC, NLI	Sentence Embeddings Similarity	One-hoc
Cover-LS 2022	CODEX, T5-large	BM25 and Diversity Selection	In Domain	SP	Diversity Training	Iterative
Prompt PG 2022a	GPT-3	Fine-tuned Retriever	In Domain	MathR	Policy Gradient	One-hoc
Dynamic Least-to-Most 2022	DODEX	Tree Structure Similarity	In Domain	SP	Diversity and Lexical Overalp	Iterative
Self-Prompting 2022	Instruct GPT, CODEX	SBERT with Clustering	Free Form Corpus	QA	Sentence Embeddings Similarity	Clustering
CEIL 2023a	GPT-NEO, GPT2-XL, CODEX	BM25, BERT, Fine-tuned Retriever	In Domain	SA, PD, NLI, CSR, QA, CodeGen, and SP	Diversity Training	Iterative
R-BM25 2023	XGLM-7.5G	BM25, BM25 and Term Recall based rerank In Domain	In Domain	MT	Term-based Similarity	One-hoc
ICL-MC 2023	GPTNeo-2.7B, XGLM-2.9G, BLOOM-3B	BM25	In Domain Cross Domain	MT	Term-based Similarity	One-hoc
ICL-ML 2023	OPT-13/175B, LLaMA-7/70B	SBERT	In Domain	Intent Classification	Sentence Embeddings Similarity	One-hoc
UP RISE 2023	GPT-Neo, BLOOM, OPT, GPT-3	Fine-tuned Retriever	Cross Domains Human Feedback	RC, QA, NLI, SA, CSR, CR, PD	Contrastive Learning	One-hoc

Paper	LLMs	Retrieval Method	Retrieval Corpus	Evaluation Tasks	Retriever Training	Retrieval Strategy
Dr.ICL 2023	PaLM, Flan-PaLM	BM25, GTR, Fine-tuned Retriever	In Domain	QA, NLI, MathR	Contrastive Learning	One-hoc
LLM-R 2023a	LLaMA-7B	Fine-tuned Retriever and Reward Model	Mix Domain	QA, CSR, CR, PD, RC, SA, D2T, Summ, NLI	Contrastive learning + KL	One-hoc
UDR 2023b	GPT-J, GPT-Neo, CODEX, GPT-3	Fine-tuned Unified Retriever with Iterative Data Mining	Mix Domain	SA, TC, CSR, NLI, SP, StoryGen, Summ, D2T	Contrastive Learning	One-hoc
MoT 2023b	ChatGPT	SBERT with Clustering and Filtering, LLM as Retriever	Unlabelled Queries with LLMs CoT	MathR, NLI, CSR, QA	Sentence Embeddings Similarity	Clustering
RetICL 2023	CODEX	Iterative LSTM with Reinforcement Learning	In Domain	MathR	PPO and GAE	Iterative
Ambig-ICL 2023	Flan-PaLM	Fine-tuned multilingual T5, and a Filtering Algorithm Based on LLM feedback	In Domain	TC, SA	Contrastive Learning	One-hoc

Table 1: Comparison with Related Work. Abbreviation for Evaluation Tasks: CodeGen (code generation), SA (sentiment analysis), Table2Text (Table to Text generation), QA (question answering), SP (semantic parsing), DST (Dialogue State Tracking), D2T (Data-to-Text), Summ (Summarization), CSR (commonsense reasoning), RC (reading comprehension), NLI (natural language inference), CR (Coreference Resolution), MathR (mathematical reasoning), PD (paraphrase detection), TQA (Table Question Answering), TC (Topic Classification), StoryGen (Story Generation), MT (Machine Translation)

2.1 In-Context Learning

As the scale of the PLMs and the scale of the datasets on which these models were pre-trained increased – leading to pre-trained Large Language Models (LLMs), it was discovered that pre-trained LLMs (hereafter, referred to as *LLMs* for brevity) have a remarkable capability of learning in-context from a few demonstrations (Brown et al., 2020). That is, LLMs were shown to be able to adapt to new tasks by only seeing a few examples of the new task in their input, as opposed to needing additional training data or fine-tuning. This is typically referred to as *few-shot in-context learning*.

Let $\mathcal{T}$ be a task and $q_* \sim \mathcal{T}$ represent a sample query from this task for which we would like to find an answer using an LLM. In the case of few-shot learning, we find or construct multiple demonstrations $\{d_1, \ldots, d_k\}$ where each demonstration $d_i = (q_i, a_i)$ contains a query $q_i \sim \mathcal{T}$ and the answer a_i to that query, and feed an input of the form

$$q_1 \; a_1 \; \ldots \; q_k \; a_k \; q_*$$

to the LLM. The input is typically referred to as *prompt*. It is common to add some separator tokens to the prompt so the boundaries of the demonstrations and the questions and answers within those demonstrations are clear. An example prompt will then be as follows:

$$\text{Demonstration 1}: \;\; \text{Query}: q_1, \;\; \text{Answer}: a_1$$

$$...$$

$$\text{Demonstration k}+1: \;\; \text{Query}: q_*, \;\; \text{Answer}:$$

Seeing the demonstrations as a few examples of the task, LLMs learn from the demonstrations in context (without any weight updates) and use a similar pattern to provide an answer to the query q_*. Few-shot learning is a remarkable capability of LLMs given that they are not trained on such data during their pre-training. While LLMs show strong few-shot learning capabilities off-the-shelve, it has been shown that warming them up by finetuning them on few-shot data from multiple tasks will further boost their few-shot learning capability (Min et al., 2022a; Chen et al., 2022; Radford et al., 2019).

Another remarkable ICL capability of LLMs is to learn from in-context instructions: finetuning LLMs on instructions from multiple tasks makes them learn to follow instructions for new tasks (Ouyang et al., 2022; Longpre et al., 2023; Zhang et al., 2023). In this case, commonly known as *instruction tuning*, the LLM is finetuned on data of the type $I^{\mathcal{T}}, q^{\mathcal{T}}, a$ where $I^{\mathcal{T}}$ represents the instructions for a task $\mathcal{T}$ describing how the task should be performed, $q^{\mathcal{T}}$ represents a query from task $\mathcal{T}$ and a represents the answer. The finetuning is performed on data from multiple tasks and multiple queries from each task. It is also possible to combine instructions with few-shot demonstrations in which case an example prompt may be as follows:

$$\text{[Task instructions]}$$
$$\text{Demonstration 1: Query: } q_1, \quad \text{Answer: } a_1$$
$$\ldots$$
$$\text{Demonstration k+1: Query: } q_*, \quad \text{Answer: }$$

Benefits of ICL: Compared to the aforementioned approach of utilizing LLMs which involves pre-training followed by fine-tuning, ICL offers several key advantages. Firstly, fine-tuning may not always be feasible due to restricted access to the LLM, inadequate computational resources, or inadequately labeled data (Brown et al., 2020), whereas ICL requires fewer resources, less data, and is easier to serve through API calls. Additionally, ICL avoids the issues commonly associated with fine-tuning, such as overfitting or shocks (Ying, 2019; Kazemi et al., 2023a), as it does not modify the model's parameters, allowing it to remain general.

2.2 What Makes for Good Demonstrations?

Several works try to provide theoretical justifications and insights into how LLMs learn from a few in-context demonstrations (Xie et al., 2021; Garg et al., 2022; Von Oswald et al., 2023). However, the exact reasons behind this capability are still largely unclear making it difficult to select optimal few-shot demonstrations. Fortunately, various empirical results show the effect of the few-shot demonstrations on the predictive accuracy of the LLMs and provide suggestions on the best practices for preparing them. They also show the brittleness of the LLMs in the choice, format, and order of the few-shot demonstrations. Here, we describe some of the more prominent ones.

Number of Demonstrations: LLMs generally benefit from more demonstrations, but as the number of demonstrations increases the rate of improvement typically decreases (Brown et al., 2020; Ye et al., 2023b; Min et al., 2022b). Generation tasks have been shown to benefit from an increased number of demonstrations more than classification tasks (Li et al., 2023b). Toward increasing the number of demonstrations, one barrier is the maximum context size of the LLM. While the size of the context has been increasing over time with newer LLMs, it may still be problematic for datasets with long input texts or classification datasets with many classes.

Demonstration Formatting: Various works have shown that the formatting and wording of the prompts can play a crucial role in the performance of the LLM (Jiang et al., 2020; Shin et al., 2020; Kojima et al.; Yang et al., 2023). For example, Kojima et al. show that simply adding *Let's think step by step* to the prompt makes LLMs reason step by step and solve substantially more problems, and Weller et al. (2023) show that adding *According to Wikipedia* to the prompt makes them more factual. Moreover, Min et al. (2022b) shows that besides the text formatting, the label space and the distribution of the input text in the demonstrations are also of immense importance.

Order of Demonstrations: The order of demonstrations has been shown to substantially affect the model performance. For example, Lu et al. (2022b) show that on some tasks, the model performance can range from near-random to state-of-the-art depending on the order of the prompts, and Zhao et al. (2021) show that answers appearing toward the end of the prompt are more likely to be predicted by the model.

Diversity of Demonstrations: Another important factor in the success of few-shot learning is the diversity of the demonstrations. Naik et al. (2023) propose *DiversePrompting* where for the question of a demonstration, an LLM is used to generate different ways of solving the problem, and then those solutions are used in the prompt. Zhang et al. (2022b) propose to select a diverse set of questions as few-shot examples. Ma et al. (2023) propose a fairness metric for selecting demonstrations which encourages selecting diverse few-shot demonstrations that produce a near uniform predictive distribution for a semantic-free input.

Chain of Thought (CoT): It has been shown that including a rationale for the answer significantly improves model performance, especially for models that are larger than a certain size (Suzgun et al., 2022). The rationale is commonly known as *chain of thought (CoT)* (Wei et al., 2022). In the case of CoT prompting, the demonstrations are typically formatted as:

$$\text{Query} : q_i, \quad \text{Rationale} : r_i, \quad \text{Answer} : a_i$$

with the rationale appearing before the final answer. Several works have investigated the reason behind the efficacy of CoT prompting and how to improve the prompts and rationales (Wang et al., 2022a; Lanham et al., 2023).

3 In-context Learning with Demonstration Retrieval

Traditionally, the same set of few-shot demonstrations is used on all queries, which can be suboptimal especially when there are high variations among the queries. An alternative is to *retrieve* few-shot demonstrations that are tailored to the current query. Previous work has shown that demonstration retrieval leads to substantial improvements in the task metrics, compared to manually curated or randomly selected demonstrations (Luo et al., 2023; Ye et al., 2023a). Furthermore, LLMs have been shown to become less sensitive to the factors such as demonstration ordering (Section 2.2) when retrieved demonstrations are used (Li et al., 2023b).

This section gives an overview of the retrieval-based ICL (RetICL). We start by defining ICL with *retrieved demonstrations*. Formally, given a query q_* and a **retrieval corpus** $\mathcal{C}$, a **demonstration retriever** $\mathcal{DR}$ selects a set of demonstrations $\{d_1, \dots, d_k\} \sim \mathcal{C}$, where each demonstration is $d_i = (q_i, a_i)$. The LLM input sequence becomes $(d_1, \dots, d_k, q_*)$. The goal of the retriever is to select demonstrations that maximize the probability of the correct answer a_*.

The success of RetICL depends on several factors. This section explores design choices, including the retrieval objectives, retrieval inference strategy, and retrieval corpus. Then in Sections 4 and 5, we explore the retriever models and how to train them to tailor to downstream tasks.

3.1 Retrieval Objectives: Similarity and Diversity

Various retrieval objectives for selecting and tailoring in-context examples for LLMs have been explored (Luo et al., 2023; Rubin et al., 2022; Ye et al., 2023a; Dalvi et al., 2022; Cheng et al., 2023; Li et al., 2023b). There are two primary retrieval objectives for selecting demonstrations: similarity and diversity. Similarity involves selecting demonstrations most akin to the query and can be based on language similarity (term matching or semantic matching), structural aspects (sentence structure, reasoning structure, etc.), or other criteria. Most studies focus on language similarity, with fewer addressing structural similarity, often due to the challenges in extracting a query's structure in many tasks (Levy et al., 2022). Beyond similarity, some work has found that the diversity of demonstrations is important. The motivations for diversity include avoiding repetitive demonstrations (Zhang et al., 2022b), bringing different perspectives (Yu et al., 2023), and maximizing the demonstrations' coverage of the test query, in terms of covering either its words or syntactic structures (Levy et al., 2022). Measuring the diversity of multiple demonstrations is a major technical challenge. Ye et al. (2023a) applied determinantal point processes (DPP) a probabilistic model to measure the negative interaction (Kulesza et al., 2012), to measure the diversity. Levy et al. (2022) found that diversity and coverage are important when the model is unfamiliar with the output symbols space. It is noteworthy that researchers have found that ICL benefits more from demonstrations with higher complexity in some scenarios (Fu et al., 2022), where they define the complexity in terms of the query length or reasoning steps. However, Fu et al. (2022) employed heuristic rules to define complexity and pre-selected demonstrations accordingly. Their research revealed that using a similarity-based retriever led to improved performance in a specific mathematical reasoning task. This might indicate that combining similarity and complexity considerations could be a promising strategy for enhancing the approach to reasoning tasks.

3.2 Inference Strategy to Retrieve Few-shots Demonstrations

This section explores various strategies for employing a retriever to gather k demonstrations. We divide these into three distinct methodologies.

One-hoc Retrieval This is the most basic retrieval strategy. To obtain k demonstrations, given a query, the retriever ranks the demonstrations based on some scoring criteria and then selects the top-k demonstrations. Thus, each demonstration is chosen independently of the others. This method is straightforward and fast, however, it might not yield the best combination of k demonstrations as these demonstrations might be homogeneous.

Clustering Retrieval To mitigate the issue of homogeneity in one-hot retrieval, clustering retrieval approaches (Li et al., 2022; Zhang et al., 2022b; Li & Qiu, 2023b) categorize all demonstrations into k sub-groups aiming to group similar demonstrations together. Then given a query, the retriever picks the most similar demonstration from each sub-group resulting in a final set of k demonstrations. The core principle of clustering is to select a diverse range of demonstrations. Most of the work use SBERT Reimers & Gurevych (2019a) to encode the demonstrations (only the question or the entire demonstrations) and then apply k-means for clustering.

Iterative Retrieval The earlier retrieval strategies acquire each demonstration independently. However, in iterative retrieval, a retriever selects demonstrations based on both the query and previously retrieved demonstrations. This process starts with a single query, for which the retriever finds one best demonstration. The query is then augmented (e.g. combined with the demonstration) to retrieve the next demonstration. This step is iteratively executed k times to gather k demonstrations. The general idea is to select the demonstrations that can complement each other. An an example of a work from this categorym, Scarlatos & Lan (2023) train an LSTM retriever using a reinforcement learning framework. During the inference phase, the retriever processes the input query to select the best initial demonstration. It then generates a new query representation by integrating the query with prior demonstrations, specifically utilizing the hidden state representation from the LSTM model. This process of updating the query representation and obtaining subsequent demonstrations continues iteratively until k demonstrations are retrieved.

3.3 Retrieval Corpus

The retrieval corpus forms a pool of demonstrations that the retriever can access. Using annotated data is one of the most straightforward ways to construct the retrieval corpus. This setting assumes that training data related to a task is available, and thus can be used as the retrieval corpus. Under this setting, there are three main ways to construct the corpus that we will discuss individually below.

In-Domain In this setting, an in-domain training set, independently and identically distribution (IID) with the test queries, is available and serves as the retrieval corpus. Most existing work take the full training set as the corpus. However, to be more annotation efficient, Hongjin et al. (2022) uses only a subset M of the training set N which includes the most representative and diverse ones, where $|M| << |N|$. One question that remains unanswered from the work of Hongjin et al. (2022) is how the predictive performance is affected as a function of retrieving from a subset M instead of the entire training set N. While there is no follow-up work to answer this question, the closest comparison we find is the results in Ye et al. (2023a) where a similar setup as Hongjin et al. (2022) is used except that they use the entire training set as the retrieval corpus, and report lower performance on the SST-5 dataset (compare the Figure 3 in Hongjin et al. (2022) and Table 3 in (Ye et al., 2023a)). While there might be other differences between the two setups that may affect the final performance, this comparison implies that retrieving from a carefully selected subset might have comparable results to retrieving from the entire training set.

Mix-Domain The previous scenario has one individual retrieval corpus for different tasks. Assuming that we want to test model performance on two tasks, then in the in-domain setting, there will be two retrieval corpora separately. Furthermore, the in-domain setting assumes that the model has knowledge about which task the test question belongs to such that when it comes to the retrieval phase, it knows which corpus to select the demonstrations from. However, this assumption does not hold in several real-world applications of LLMs. In the mix-domain setting (Wang et al., 2023a; Li et al., 2023b), the retrieval corpus is constructed from the combination of all tasks. At the inference time, given a question, the retriever will retrieve demonstrations from this mixed corpus; the demonstrations can come from the same domain as the test question or from other tasks.

Cross-Domain In this setting, IID human-annotated demonstrations are not available for the test queries, so one uses annotated demonstrations from other similar tasks (Cheng et al., 2023; Shi et al., 2022). Note that this is different from the mix-domain setting where part of the corpus is IID and part of it is not. For instance, Shi et al. (2022) describes a scenario where the goal is to parse a Chinese query into SQL. However, the demonstrations are sourced from an English Text-to-SQL corpus, a domain with significantly more resources than the target domain. Shi et al. (2022) employs this high-resource data as the retrieval corpus. To adapt to the target domain during inference with a LLM, the target query is translated into the same language as the demonstrations. Nie et al. (2022) presents a similar approach, retrieving demonstrations from high-resource domains to address low-resource queries. However, their retrieval pool consists of multiple high-resource sources.

Unlabelled Queries with Automatically Generated Answers The previous three corpora all presuppose the availability of human-annotated data. However, this assumption may not hold in real-life scenarios, particularly in streaming settings where users can pose questions without any pre-annotated answers. Several studies (Zhang et al., 2022b; Li & Qiu, 2023b) have suggested using LLMs to generate answers for unlabeled data. They apply filtering techniques to determine the quality of these generated answers, adding only those examples with high-quality answers to the retrieval corpus. The most widely used filtering technique is based on self-consistency (Wang et al., 2022c). This approach involves prompting the language model to generate multiple chains of thought and answers, then selecting the most common answer as the final response.

Free Form Corpus Another approach to deal with the lack of human-annotated data for similar tasks is create pseudo-demonstrations from unstructured text. Toward this goal, Lyu et al. (2022) utilized the Demix dataset (Gururangan et al., 2022), which is not tailored for any specific task. To generate pusedo-demonstrations, a retriever selects the top-k most relevant sentences from the dataset. Subsequently, arbitrary labels are attached to each sentence to form the examples. Li et al. (2022) propose a synthetic question answering generation method to create QA pairs using the synthetic generated passages by an LLM.

4 Off-the-shelf Demonstration Retrievers

To achieve the retrieval objectives outlined above, researchers have explored various types of demonstration retrievers. A typical demonstration retriever encodes examples from the retrieval corpus and the query into some vector representations, and then a similarity measure (e.g. cosine similarity) is calculated between candidate demonstration embeddings and the query embedding to locate the most related demonstrations. Given the limited understanding of the underlying mechanism through which retrieved demonstrations enhance the performance of LLMs, initial research efforts focused on a heuristic evaluation of readily available retrievers for this task. Subsequent research endeavors explored the design and development of learning-based retrievers specifically customized for retrieving demonstrations. This section reviews representative off-the-shelf models and we will discuss the learning-based models in Section 5.

Term-based Similarity BM25 (Robertson et al., 2009) is one of the most popular term-based scoring methods due to its simplicity and effectiveness in producing relevant results. It takes into account both term frequencies and document lengths. It has been empirically demonstrated in various works (Luo et al., 2023; Rubin et al., 2022; Agrawal et al., 2022; Ye et al., 2023a; Dalvi et al., 2022) that using BM25 to select similar examples as few-shots in ICL can help improve the performance of many LLM inference tasks. While BM25 has become a standard baseline model in the field, it is not without its limitations. Due to its sole reliance on term frequency and document length, this approach may overlook crucial aspects such as semantic meaning and sentence structure, potentially leading to inaccuracies in certain instances. Another drawback is that BM25 lacks the capability for fine-tuning in downstream tasks, making it less competitive compared to neural models which can be fine-tuned and customized for specific downstream tasks.

Sentence Embedding Similarity In this approach, queries and documents are encoded to the same dense embedding space using an off-the-shelf sentence embedding model, and then similarity scores (e.g. cosine similarity) are calculated to rank the most relevant documents for each query. A rich collection of sentence embedding methodologies exists in the literature. Sentence-BERT (SBERT) (Reimers & Gurevych, 2019a) is a modification of the pretrained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. The effectiveness of SBERT embeddings for demonstration retrieval has been investigated in several works (Rubin et al., 2022; Li & Qiu, 2023b; Wang et al., 2023a), and the results show that retrieving demonstrations based on SBERT embeddings often provides a boost in performance compared to zero-shot or random few-shot selection. In the KATE method (Liu et al., 2022), the authors studied using vanilla RoBERTa (Liu et al., 2019) and finetuned RoBERTa on NLI (Bowman et al., 2015b) and STS-B (Cer et al., 2017) datasets for selecting good demonstrations, and found that the finetuned version on task-related datasets offered further empirical gains. Note that here, the demonstration retriever is not trained for ICL demonstration retrieval based on task-specific data (a topic which we will discuss in Section 5); instead, the retriever is finetuned related tasks to provide a better notion of similarity for the task at hand. So we still categorize it as an off-the-shelf retriever. Shi et al. (2022) extends the use case to cross-lingual few-shot retrieval in the Text to-SQL semantic parsing task, and they use mSBERT (Reimers & Gurevych, 2019b), mUSE (Yang et al., 2019) and mT5 (Xue et al., 2020) as the baseline models for comparison. Other widely used baseline models for demonstration retrieval include $E5_{\text{base}}$ (Wang et al., 2022b), SimCSE (Gao et al., 2021b). Instead of relying on "word matches" as in BM25, these sentence embedding similarity approaches can better capture semantic similarity (for example, synonyms, and related topics), however computationally they might be more expensive.

Pretrained Dual Encoder In the context of demonstration retrieval where the goal is to identify relevant examples for a given query, the query is typically a question, while the examples may contain additional information such as answers, chains of thoughts, supporting knowledge, or even follow different patterns. Therefore, transforming them into a uniform embedding space to calculate relevance might not be the most effective approach. In this case, LLM retrieval architectures such as Dual Encoder that are pretrained on retrieval or question-answering tasks can better grasp the intricate relationships between complex logical concepts and reasoning processes by employing different semantic embeddings for queries and candidates (Li & Qiu, 2023b). In practice, training a dual-encoder can be highly expensive as it typically requires a large training corpus. Fortunately, there are publicly available pretrained retrievers, although not specifically optimized for few-shot retrieval tasks, already demonstrating success in helping LLMs to learn from the selected examples. Luo et al. (2023) studied applying GTR (Ni et al., 2021) to select semantically similar examples as demonstrations, and empirically proved that this approach brought in better performance gain than random fewshots for both PaLM (Chowdhery et al., 2023) and FLAN (Chung et al., 2022) models. GTR is a T5-based dual encoder model that is pretrained on the CommunityQA (Abujabal et al., 2019) and finetuned on the MS Marco dataset (Nguyen et al., 2016). Moreover, Khattab et al. (2022) reported results for employing ColBERTv2 (Santhanam et al., 2021) as the retrieval module in their DEMONSTRATE–SEARCH–PREDICT (DSP) framework for ICL. ColBERTv2 is a state-of-art retrieval model that adopts the late interaction architecture (Khattab & Zaharia, 2020) and is trained on the MS Marco dataset. In the proposed framework, it is used to retrieve both (i) related knowledge during the search stage and (2) top k similar examples as demonstrations.

5 Fine-tuned Demonstrations Retrievers

Although off-the-shelf retrievers have shown some promise in retrieving demonstrations for LLMs, the retrieved demonstrations given by the off-the-shelf retrievers might not represent the nature of the task and how the task should be solved in general. Therefore, it might lead to sub-optimal performance. Researchers thus have started to explore learning-based methods to further push the boundaries. A typical objective when designing a good demonstration retriever is: if an LLM finds a demonstration useful when being used as an illustrative example, the retriever should be encouraged to rank the demonstration higher. This allows us to train models directly relying on signals from query and output pairs in the task of interest, without human annotations. To develop a demonstration retriever, the majority of approaches utilize current dual encoder models (Karpukhin et al., 2020; Ni et al., 2021). The key variations lie in the methods of gathering training data and formulating training objectives. We will explore these aspects in more detail in the subsequent sections.

5.1 Collecting Training Data for Demonstration Retriever

Based on LLMs Signals A popular approach to collecting training examples is to use the supervisory signals from LLMs. In this case, a typical paradigm is to first employ some filtering mechanisms (Cheng et al., 2023) or unsupervised retrievers (e.g. BM25 and SBERT) (Luo et al., 2023) as the initial retriever, this step can help limit the pool size for mining the right training data. Then a scoring LLM, which serves as a proxy for the inference LLM, is used to score each candidate demonstration d. Here the score is defined as $s(e) = p(a|d, q)$ which is the conditional probability of output answer a given the input query q and demonstration d. Another approach is to train a smaller reward model that can provide more fine-grained supervision for dense retrievers. For example, Wang et al. (2023a) proposed to finetune a cross-encoder model serving as a teacher model for training the retriever.

Once a score is obtained, a retriever can be trained that predicts these scores directly (Ye et al., 2023a). Alternatively, the candidate demonstrations can be ranked for each query based on their scores, considering the top-ranked demonstrations as *positive* examples that help the LLM get to the right answer and the bottom-ranked ones as *negative* examples that mislead the LLM towards the wrong answers; then a retriever can be trained which separates positive examples from negative examples (Rubin et al., 2022; Cheng et al., 2023; Luo et al., 2023).

There are different strategies for choosing the scoring LLM. Ideally, one uses the inference LLM itself as the scorer in order to perfectly reflect its preferences (Li et al., 2023b; Shi et al., 2022). However, training retrievers requires large amounts of labeled data, and it may be expensive use very large models for labeling. Consequently, for scoring one may gravitate towards utilizing smaller models, especially those within the same model family as the inference LLM (Luo et al., 2023; Cheng et al., 2023; Rubin et al., 2022).

Model-Free One approach to collecting training data for demonstration retriever is to directly measure the similarity between the labels of the candidate demonstrations and the label of the query, and use this similarity as a proxy of the importance of a demonstration (Hu et al., 2022; Poesia et al., 2021). For instance, Hu et al. (2022) explored a dialogue context where labels are structured as a sequence of stages. The similarity between a query's label and a demonstration's label is determined by calculating the average F1 scores of these two labels. This method adopts a heuristic approach (i.e. stage changes), presuming that the similarity metric can closely resemble the preference for good demonstrations from an LLM, and it often necessitates domain-specific expertise for design.

5.2 Training Objectives

Thus far, we have explored the creation of training data for demonstration retrievers in the context of ICL. We now proceed to examine the commonly used loss functions for training retrievers.

List-wise Ranking Loss The list-wise ranking approach looks at a list of candidate documents for a given query and tries to capture the correct ordering for it. Li et al. (2023b) proposed to inject the ranking signals into the retriever using an approach inspired by LambdaRank (Burges, 2010). More formally, given each query q, they first rank all l candidate documents according to their relevant scores $S = \{s(d_i)\}_{i=1}^{l}$, according to which the associated ranking $\mathcal{R} = \{r(d_i)\}_{i=1}^{l}$ is computed. Then the loss function is defined as follows:

$$\mathcal{L}_{\text{listwise}} = \sum_{d_i, d_j} \max\left(0, \frac{1}{r(d_i)} - \frac{1}{r(d_j)}\right) * \log(1 + e^{\text{sim}(q_i, d_j) - \text{sim}(q_i, d_i)})$$

where $\text{sim}(q, d)$ is the relavance between a candidate demonstration d and the input q. In the list-wise ranking objective, retriever can benefit from the full ranking of the candidate set to make accurate predictions for the most relevant demonstrations. However, obtaining the full ranking list and calculating the loss function on top of it might be very expensive and time-consuming. Additionally, the model is trained to discern the relative preferences between examples without explicitly determining whether an example can serve as an absolute good demonstration.

InfoNCE Loss Another widely adopted training procedure is contrastive learning using the InfoNCE loss (Rubin et al., 2022; Cheng et al., 2023; Luo et al., 2023). When positive and negative examples can be correctly identified, InfoNCE loss is an effective loss function because it can take advantage of the supervisory labels to produce a representation that sets apart the useful examples for demonstration retrieval. In this approach, each training instance is given in the form of $< q_i, d_i^+, d_{i,1}^-, ...d_{i,k}^- >$. Here d_i^+ is a selected positive example concerning the input q_i, and the negative examples consist of one hard negative example $d_{i,1}^-$ and k random examples from the other instances in the same mini-batch. Then the typical contrastive loss can be defined as

$$\mathcal{L}_{\text{cont}} = \mathcal{L}(q_i, d_i^+, d_{i,1}^-, ...d_{i,k}^-) = -\log \frac{e^{\text{sim}(q_i, d_i^+)}}{e^{\text{sim}(q_i, d_i^+)} + \sum_{j=1}^{k} e^{\text{sim}(q_i, d_{i,j}^-)}}$$

The random negative examples from the same mini-batch are called *in-batch negatives.* They are typically selected from both the positive examples and hard negative examples of other instances.

Distillation by KL Divergence Ye et al. (2023a) claims that although the InfoNCE loss has been found effective in training demonstration retrievers and can learn which examples might be superior to others, it has the same treatment for all negative examples and the predicted scores from LLM are not fully utilized. As an alternative to train a demonstration retriever using positive and negative examples, Shi et al. (2022) proposed to train the retriever by directly distilling the LLM's scoring function. More specifically, the retriever model is designed to produce ranking scores that match the usefulness of a demonstration to help with the LLM inference; this is done by minimizing the KL-divergence between the top K examples score distribution from scoring LLM and the ranking score distribution produced by the retriever

$$\mathcal{L}_{\text{distill}} = \text{KL}(p_{\text{LLM}} || p_{\text{retriever}}) = \sum_{k=1}^{K} p_{\text{LLM}}(d_k) log\left(\frac{p_{\text{LLM}}(d_k)}{p_{\text{retriever}}(d_k)}\right)$$

Multiple Objectives In Wang et al. (2023a), the authors proposed to train the demonstration retriever model with combined objectives: (1) knowledge distillation from the trained reward model which can capture the preferences of LLMs over the retrieved candidates (2) InfoNCE-based contrastive loss to incorporate the in-batch negatives. More specifically, the resulting loss function is as follows:

$$\mathcal{L}_{\text{combined}} = \alpha \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{distill}}$$

Here α is a constant that controls the relative importance of the two losses. They claimed that with the multi-objective function, both the absolute scores and supervised signals are taken into consideration. Li et al. (2023b) trains a universal retriever with both list-wise ranking loss and the InfoCNE loss.

Iterative Training Regarding training strategies, most research efforts have centered on fine-tuning a single retriever. Wang et al. (2023a) and Li et al. (2023b) instead proposed to iterate the retriever model multiple times. More specifically, the retriever trained in iteration i will be employed to retrieve a new set of candidates for the subsequent iteration $i + 1$. Such an iterative training approach allows progressively improving retriever quality by mining better positive and hard negative examples at each iteration.

Diversity Training The Determinantal Point Process model (Alex Kulesz, 2012) defines a probability distribution over all the combinations of candidate demonstrations, giving high probability to subsets that contain relevant and diverse items (Levy et al., 2022). It models diversity by incorporating cross-candidate similarity scores, and models similarity via a per-candidate relevance score, i.e., a similarity score between a candidate and the test query. In addition to using DPP directly (Levy et al., 2022), Ye et al. (2023a) also fine-tuned a DPP model and demonstrated meaningful improvements over pure similarity-based methods.

Re-ranker Training It is not uncommon that people adopt a two-stage retriever-reranker architecture for ICL retrieval in the literature to further improve the exemplar selection process (Shi et al., 2022). Generally, a dual-encoder-based retriever can encode query and candidate documents for fast indexing and searching, but neglect the finer-grained token-level interactions. Cross-encoder-based reranker, on the other hand, can capture the subtle relationship but is time-consuming. We can benefit from both of these methods by chaining two methods together. In the first stage, a retriever model is used to quickly select the top N examplers to limit the candidate pool of interest, then a reranker reranks the retrieved N examplars and uses the top K exemplars to construct a prompt. Sigmoid cross-entropy loss is typically used for training the reranker. Lu et al. (2022a) also utilizes a similar structure as the reranker to select the demonstrations from random candidates. The reranker is trained using reinforcement learning.

5.3 Summary

Here, we summarize the advantages and disadvantages of various retriever models. The off-the-shelf retrievers are easy to use without any downstream task finetuning and typically demonstrate stronger performance than random demonstrations. One exception is in commonsense reasoning tasks where Zhang et al. (2022b) and Ye et al. (2023a) found that for these tasks, random demonstrations are consistently better than retrieval-based method. Cheng et al. (2023) also show that retrieved demonstrations harm commonsense reasoning and coreference resolution tasks. Among the three categories of off-the-shelf retrievers, sparse retrievers such as BM25 are more index-efficient. This feature becomes particularly valuable when dealing with large volumes of demonstrations and limited hardware memory, making BM25 a preferable choice under such circumstances. In contrast, sentence-embedding similarity-based methods and dual-encoder-based retrieval systems, which are trained on language tasks, excel in capturing more semantically focused retrieval. Regarding performance, Luo et al. (2023) compared BM25 with dual encoder (GTR) across 5 tasks, and they found that the average performance of these two is very similar (within 0.5% difference), and BM25 outperformed the dual encoder in some tasks and vice versa. In another study, Ye et al. (2023a) observed a similar trend highlighting that no single retriever consistently outperforms others across different tasks. Both Rubin et al. (2022) and Li et al. (2023b) found that BM25 is better than SBERT on semantic parsing tasks, while Li et al. (2023b) found that SBERT is better than BM25 on sentiment analysis tasks. Nevertheless, retrievers that are fine-tuned demonstrate superior performance compared to their off-the-shelf counterparts. The main drawback of fine-tuned retrievers lies in the high cost of obtaining training data. Additionally, the common practice of employing task-specific retrievers complicates the system and limits its generalizability. Li et al. (2023b) proposed to train a universal retriever that shows stronger performance than task-specific demonstration retriever (e.g. EPR (Rubin et al., 2022)) on most of the tasks.

6 Applications

The effectiveness of retrieval-based ICL has been showed in four categories of tasks. 1). natural language understanding, 2- reasoning, 3- knowledge-based QA, and 4- Text generation. We discuss each category below.

Natural language understanding tasks that benefit from RetICL include sentiment analysis (SA) (Socher et al., 2013; Zhang et al., 2015; Go et al., 2009), paraphrase detection (PD) (Dolan et al., 2004; Zhang et al., 2019), reading comprehension (RC) (Rajpurkar et al., 2016; Khashabi et al., 2018; Clark et al., 2019; Khashabi et al., 2018; Clark et al., 2019; Mihaylov et al., 2018), and natural language inference (NLI) (Williams et al., 2018; Wang et al., 2018; Bowman et al., 2015a; De Marneffe et al., 2008). Specially, RetICL shows noticeable improvements on SA and NLI tasks (Liu et al., 2022; Ye et al., 2023a).

Reasoning tasks tasks that benefit from RetICL include mathematical reasoning (Cobbe et al., 2021; Lu et al., 2022a; Ling et al., 2017), commonsense reasoning (CSR) (Talmor et al., 2019; Zellers et al., 2019; Bisk et al., 2020; Roemmele et al., 2011),and ethical Reasoning (Jiang et al., 2021). Such tasks are usually accompanied by CoT, but Zhang et al. (2022b) found that CoT does not help that much for the commonsense reasoning task. Many works have shown that retrieval-based methods are worse than random demonstrations on commonsense reasoning tasks (e.g. CMSQA). A simple similarity-based retrieval method does not show significant improvement in mathematical reasoning tasks, and Zhang et al. (2022b) shows that diversity is important for mathematical reasoning tasks. The iterative retrieval strategy shows the most significant improvement on mathematical reasoning tasks Scarlatos & Lan (2023).

In Knowledge-based QA, external knowledge is required to answer the question (Berant et al., 2013; Kwiatkowski et al., 2019; Joshi et al., 2017; Clark et al., 2018). To tackle such tasks, the state-of-the-art systems usually retrieve relevant passages that might contain the answer to the question, and then feed such passages and questions together to a language model to generate the answer. Liu et al. (2022) shows that using retrieval-based ICL (sentence semantic similarity-based retriever with GPT-3) is almost comparable to a fine-tuned method. Ye et al. (2023a) shows that BM25 achieve 10+% improvement on open-domain QA.

Text generation tasks that benefit from RetICL includes code generation (CodeGen) (Zelle & Mooney, 1996; Lin et al., 2018), semantic parsing (SP) (Wolfson et al., 2020; Li et al., 2021; Andreas et al., 2020), text-to-SQL (Shi et al., 2022), Table-to-text (Table2Text) generation (Parikh et al., 2020); Data-to-Text (D2T) (Nan et al., 2021; Dušek et al., 2019). Rubin et al. (2022) shows that the retrieved demonstrations significantly outperform random demonstrations (e.g. BM25 is 25+% better than random, and EPR is 30% better than random).

Apart from different types of tasks, Hongjin et al. (2022) shows that in scenarios with limited training data, RetICL outperforms fine-tuning a model on such sparse data. Furthermore, leveraging data from a high-resource domain can enhance performance in a low-resource domain, as seen in cross-lingual contexts (Shi et al., 2022; Nie et al., 2022; Cheng et al., 2023).

7 Discussion of Future Direction

Retrieve Demonstrations From Raw Text Much research assumes the availability of annotated samples that can be utilized as a retrieval corpus. Yet, when faced with a novel task, it is often the case that no such training dataset exists. While there are preliminary efforts to create pseudo demonstrations from open-ended corpora like Wikipedia (Lyu et al., 2022), the proposed method is restricted to classification tasks and the label to the demonstrations are randomly assigned. A potential approach to obtain pseudo demonstrations for generation tasks is Wan et al. (2023), where they assume a set of unlabelled queries available (without ground truth labels), and use LLMs to generate chain-of-thoughts and answers and then apply self-consistency (Wang et al., 2022c) to select high-quality demonstrations to form pseudo demonstrations pool. Employing this method of generating answers with sentences retrieved from a free-form corpus could potentially create high-quality pseudo demonstrations.

Choosing the Type of Retriever Another critical consideration in this domain is the selection of the type of retriever. The options range from neural model retrievers and sparse retrievers to template-based retrievers. The objectives range from similarity and diversity to complexity. Current research implies that no single type has emerged as universally superior. This leads to an important open question: Is there a potential for a specific type of retriever to consistently yield superior performance across a variety of tasks? Investigating this will be a key direction for future research.

Retriever Training Methods In Section 5, we explore various methods for training a retriever to search demonstrations. These methods largely depend on using an LLM to identify positive and negative demonstrations for a given question. This approach, while innovative, comes with significant computational demands. Moreover, the ambiguity in choosing what constitutes a positive or negative demonstration raises concerns about the quality of the training data for the retriever (Hashimoto et al., 2023). Addressing these challenges is crucial for the development of more efficient and reliable retriever training methods.

Active Demonstration Retrieval Much of the current research is based on a static framework where the retrieval corpus remains constant. In practical situations, input distributions may change over time and models may come across new instances. In these cases, one may like to keep updating the retrieval corpus based on the new incoming queries, but since the labels for the new samples may not be available, a selection strategy is needed to select a representative subset of the incoming examples for annotation so they can be added to the retrieval corpus. This problem is reminiscent of the well-studied active learning problem. Zhang et al. (2022a) have studied how to actively select examples with unlabelled data, however, this study is not based on the retrieval setting. The combination of active learning and RetICL can be an interesting future direction.

Retrieved Demonstrations for Small LM Most of the existing work focus on LLMs (more than 100 B parameters). Research into small LMs has recently received more attention due to its inference efficiency. Examining and improving the ICL capabilities of small LMs by better (ideally optimal) demonstration selection is an interesting direction to explore.

Theoretical Understanding of Why Are Similar Demonstrations Better Demonstrations? While lots of research has shown that similar demonstrations are better than random, it is still unknown why similar demonstrations are helpful. Ye et al. (2023a) found that the generation task is more beneficial compared to the classification task, and one possible reason is that the retrieved demonstrations could have similar answers to the input question and the model might just copy the answers. However, such an explanation does not illustrate why retrieved demonstrations are better than random ones on classification tasks. There are some hypotheses that similar demonstrations help locate the knowledge in the LLMs or based on the hypothesis of LLM conduct implicit gradient descent, then the similar demonstrations provide a more useful training signal to the input query. The research on why ICL works can help understand why similar demonstrations are better than random ones.

Fine-tuning For RetICL Few-shot Learning Most of the existing research utilizes a frozen LLM as the few-shot learner in the RetICL framework. As demonstrated by Gao et al. (2021a); Izacard et al. (2023), fine-tuning LLMs on a few-shot learning task can be a promising approach to enhancing LLM performance during inference. An intriguing avenue could involve adapting the fine-tuning strategy from open-domain question answering models to RetICL (Lewis et al., 2020; Guu et al., 2020).

RetICL for Vision and Language Models Vision and language models (VL) have demonstrated proficiency as few-shot learners (Alayrac et al., 2022; Awadalla et al., 2023; Li et al., 2023a), and researchers have integrated retrieval augmented generation methods with VL models (Luo et al., 2021; Gao et al., 2022; Yasunaga et al., 2023). In lieu of employing random demonstrations, Yang et al. (2024) leverages a retriever to select demonstrations for image captioning tasks. Additionally, Peng et al. (2023) propose an ICD-LM to generate demonstrations for a given query, with each demonstration being represented by a token. The effectiveness of this method is evidenced by its performance on image captioning and vision question answering tasks. Despite the increasing significance of vision and language generation models in real-world applications, research on RetICL for VL models remains relatively underexplored.

8 Conclusion

This survey concentrates on few-shots In-Context Learning (ICL) using retrieved examples for large language models, a key aspect of Retrieval-Augmented Generation (RAG). We outline various retrieval strategies, diverse retrieval models, retrieval pools, techniques for training demonstration retrievers, and applications. Based on the comprehensive understanding of current trends, we suggest several promising future paths for enhancing the efficacy and functionality of this approach.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. 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. 307–317, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1027. URL `https://aclanthology.org/N19-1027`.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*, 2022.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8857–8873, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Ben Taskar Alex Kulesz. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5:123–286, 2012.

Jacob Andreas, John Bufe, David Burkett, Charles Chen Jr, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, et al. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, 2015a.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 632–642, 2015b.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Christopher J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, Microsoft Research Technical Report MSR-TR-2010-82, 2010.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055, 2017. URL `http://arxiv.org/abs/1708.00055`.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. Improving in-context few-shot learning via self-supervised training. *arXiv preprint arXiv:2205.01703*, 2022.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Denvy Deng, and Qi Zhang. Uprise: Universal prompt retrieval for improving zero-shot evaluation. *arXiv preprint arXiv:2303.08518*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. URL `https://api.semanticscholar.org/CorpusID:3922816`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Bhavana Dalvi, Oyvind Tafjord, and Peter Clark. Towards teachable reasoning systems: Using a dynamic memory of user feedback for continual system improvement. *arXiv preprint arXiv:2204.13074*, 2022.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. Finding contradictions in text. In *Proceedings of acl-08: Hlt*, pp. 1039–1047, 2008.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, pp. 350–es, 2004.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, et al. Parade: Passage ranking using demonstrations with large language models. *arXiv preprint arXiv:2310.14408*, 2023.

Ondřej Dušek, David M Howcroft, and Verena Rieser. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 421–426, 2019.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5067–5077, June 2022.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. Ambiguity-aware in-context learning with large language models. *arXiv preprint arXiv:2309.07900*, 2023.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. 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. 3816–3830, 2021a.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, November 2021b.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10136–10148, 2023.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. Demix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5557–5576, 2022.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. Take one step at a time to know incremental utility of demonstration: An analysis on reranking for few-shot in-context learning. *arXiv preprint arXiv:2311.09619*, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2022.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2627–2643, 2022.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

Liwei Jiang, Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny Liang, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jon Borchardt, Saadia Gabriel, et al. Can machines learn morality? the delphi experiment. *arXiv preprint arXiv:2110.07574*, 2021.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

James H. Jurafsky, Dan; Martin. N-gram language models. In *Speech and Language Processing (3rd ed.)*. 2021.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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, 2020.

Mehran Kazemi, Sid Mittal, and Deepak Ramachandran. Understanding finetuning for factual knowledge extraction from language models. *arXiv preprint arXiv:2301.11293*, 2023a.

Seyed Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. Lambada: Backward chaining for automated reasoning in natural language. In *ACL*, 2023b.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. *CoRR*, abs/2004.12832, 2020. URL `https://arxiv.org/abs/2004.12832`.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*, 2022.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2022. *URL https://arxiv. org/abs/2205.11916*.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*, 2022.

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

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023a.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2950–2962, 2021.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. Self-prompting large language models for zero-shot open-domain qa. *arXiv preprint arXiv:2212.08635*, 2022.

Xiaonan Li and Xipeng Qiu. Finding supporting examples for in-context learning. *arXiv preprint arXiv:2302.13539*, 2023a.

Xiaonan Li and Xipeng Qiu. Mot: Pre-thinking and recalling enable chatgpt to self-improve with memory-of-thoughts. *arXiv preprint arXiv:2305.05181*, 2023b.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320*, 2023b.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *arXiv preprint arXiv:2005.00683*, 2020.

Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D Ernst. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, 2017.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*, 2022a.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022b.

Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6417–6431, 2021.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*, 2023.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. *arXiv preprint arXiv:2212.09865*, 2022.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. Fairness-guided few-shot prompting for large language models. *arXiv preprint arXiv:2303.13217*, 2023.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*, 2022.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pp. 173–184, 2023.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, 2022a.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022b.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. Diversity of thought improves reasoning abilities of large language models. *arXiv preprint arXiv:2310.07088*, 2023.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 432–447, 2021.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660, 2016.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*, 2021.

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. Cross-lingual retrieval augmented prompt for low-resource languages. *ArXiv*, abs/2212.09651, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1173–1186, 2020.

Yingzhe Peng, Xu Yang, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. Icd-lm: Configuring vision-language in-context demonstrations by language modeling. *arXiv preprint arXiv:2312.10104*, 2023.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. In *International Conference on Learning Representations*, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982—3992, 2019a.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019b.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, 2022.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *CoRR*, abs/2112.01488, 2021. URL `https://arxiv.org/abs/2112.01488`.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

Alexander Scarlatos and Andrew Lan. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*, 2023.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5248–5259, 2022.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Suzanna Sia and Kevin Duh. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pp. 173–185, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases? *arXiv preprint arXiv:2109.07154*, 2021.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 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)*, pp. 4149–4158, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan O Arik, and Tomas Pfister. Better zero-shot reasoning with self-adaptive prompting. *arXiv preprint arXiv:2305.14106*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*, 2022a.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022b.

Liang Wang, Nan Yang, and Furu Wei. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*, 2023a.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2022c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*, 2023.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2021.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020. URL `https://arxiv.org/abs/2010.11934`.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023.

Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. Exploring diverse in-context configurations for image captioning. *Advances in Neural Information Processing Systems*, 36, 2024.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernández Ábrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval. *CoRR*, abs/1907.04307, 2019. URL `http://arxiv.org/abs/1907.04307`.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-Tau Yih. Retrieval-augmented multimodal language modeling. In *International Conference on Machine Learning*, pp. 39755–39769. PMLR, 2023.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. *arXiv preprint arXiv:2302.05698*, 2023a.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. In-context instruction learning. *arXiv preprint arXiv:2302.14691*, 2023b.

Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, pp. 022022. IOP Publishing, 2019.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*, 2023.

Quan Yuan, Mehran Kazemi, Xin Xu, Isaac Noble, Vaiva Imbrasaite, and Deepak Ramachandran. Tasklama: probing the complex task understanding of language models. *arXiv preprint arXiv:2308.15299*, 2023.

John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pp. 1050–1055, 1996.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9134–9148, 2022a.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*, 2019.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2022b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.