A Survey on In-context Learning

Qingxiu Dong[1], Lei Li[1], Damai Dai[1], Ce Zheng[1], Zhiyong Wu[2], Baobao Chang[1], Xu Sun[1], Jingjing Xu[2], Lei Li[3] and Zhifang Sui[1]
[1] MOE Key Lab of Computational Linguistics, School of Computer Science, Peking University
[2] Shanghai AI Lab [3] University of California, Santa Barbara
{dqx,lilei}@stu.pku.edu.cn, wuzhiyong@pjlab.org.cn, lilei@cs.ucsb.edu
{daidamai,zce1112zslx,chbb,xusun,jingjingxu,szf}@pku.edu.cn

Abstract

With the increasing ability of large language models (LLMs), in-context learning (ICL) has become a new paradigm for natural language processing (NLP), where LLMs make predictions only based on contexts augmented with a few examples. It has been a new trend to explore ICL to evaluate and extrapolate the ability of LLMs. In this paper, we aim to survey and summarize the progress and challenges of ICL. We first present a formal definition of ICL and clarify its correlation to related studies. Then, we organize and discuss advanced techniques, including training strategies, demonstration designing strategies, as well as related analysis. Finally, we discuss the challenges of ICL and provide potential directions for further research. We hope that our work can encourage more research on uncovering how ICL works and improving ICL.

1 Introduction

With the scaling of model size and corpus size (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022), large language models (LLMs) demonstrate an in-context learning (ICL) ability, that is, learning from a few examples in the context. Many studies have shown that LLMs can perform a series of complex tasks through ICL, such as solving mathematical reasoning problems (Wei et al., 2022c). These strong abilities have been widely verified as emerging abilities for large language models (Wei et al., 2022b).

The key idea of in-context learning is to learn from analogy. Figure 1 gives an example describing how language models make decisions with ICL. First, ICL requires a few examples to form a demonstration context. These examples are usually written in natural language templates. Then, ICL concatenates a query question and a piece of demonstration context together to form a prompt, which is then fed into the language model for prediction. Different from supervised learning requiring a training stage that uses backward gradients to update model parameters, ICL does not conduct parameter updates and directly performs predictions on the pretrained language models. The model is expected to learn the pattern hidden in the demonstration and accordingly make the right prediction.

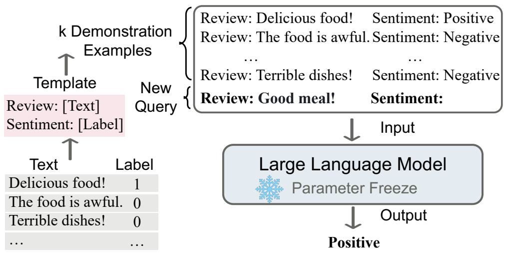

Figure 1: Illustration of in-context learning. ICL requires a piece of demonstration context containing a few examples written in natural language templates. Taking the demonstration and a query as the input, large language models are responsible for making predictions.

As a new paradigm, ICL has multiple attractive advantages. First, since the demonstration is written in natural language, it provides an interpretable interface to communicate with LLMs (Brown et al., 2020). This paradigm makes it much easier to incorporate human knowledge into LLMs by changing the demonstration and templates (Liu et al., 2022; Lu et al., 2022; Wu et al., 2022; Wei et al., 2022c). Second, in-context learning is similar to the decision process of human beings by learning from analogy (Winston, 1980). Third, compared with supervised training, ICL is a training-free learning framework. This could not only greatly reduce the computation costs for adapting the model to new tasks, but also make language-model-as-a-service (Sun et al., 2022) possible and can be easily applied to large-scale real-world tasks.

Despite being promising, there are also interesting questions and intriguing properties that re-

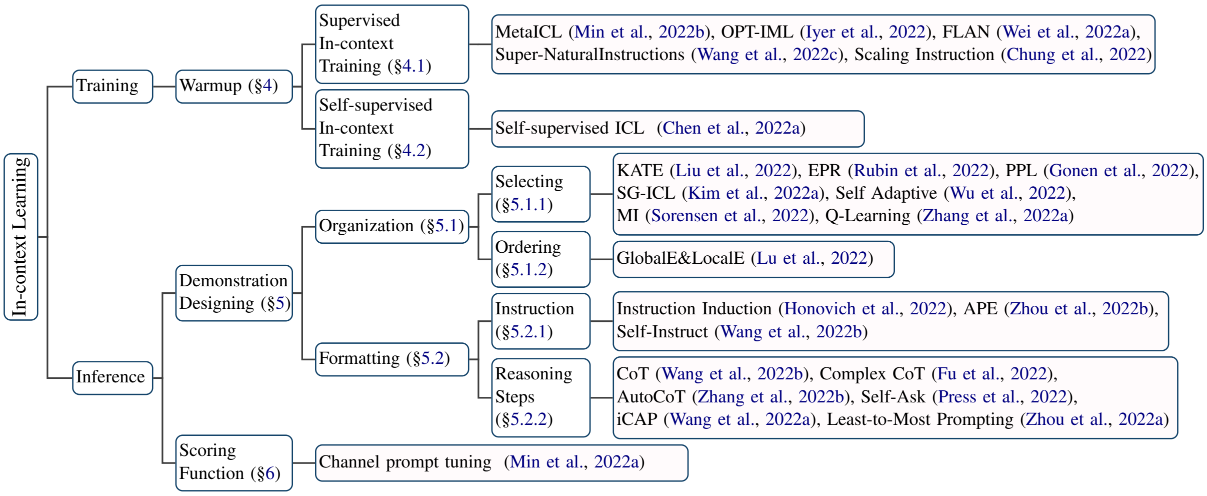

Figure 2: Taxonomy of in-context learning. The training and the inference stage are two main stages for ICL. During the training stage, existing ICL studies mainly take a pretrained LLM as backbone, and optionally warmup the model to strengthen and generalize the ICL ability. Towards the inference stage, the demonstration designing and the scoring function selecting are crucial for the ultimate performance.

quire further investigation in ICL. While the vanilla GPT-3 model itself shows promising ICL abilities, several studies observed that the ability could be significantly boosted via adaption during pretraining (Min et al., 2022b; Chen et al., 2022c). In addition, the performance of ICL is sensitive to specific settings, including the prompting template, the selection of in-context examples, and order of examples, and so on (Zhao et al., 2021). Furthermore, while intuitively reasonable, the working mechanism of the ICL remains unclear, and few studies have provided preliminary explanations (Dai et al., 2022; von Oswald et al., 2022).

With the rapid growth of studies in ICL, our survey aims to sensitize the community toward the current progress. Specifically, we present a detailed paper survey with a paper list that will be continuously updated, and make an in-depth discussion on related studies of ICL. We highlight the challenges and potential directions and hope our work may provide a useful roadmap for beginners interested in this area and shed light on future research.

2 Overview

The strong performance of ICL relies on two stages: (1) the training stage that cultivates the ICL ability of LLMs, and (2) the inference stage where LLMs predict according to task-specific demonstrations. In terms of the training stage, LLMs are directly trained on language modeling objectives, such as left-to-right generation. Although the models are not specifically optimized for in-context learning, they still exhibit the ICL ability. Existing studies on ICL basically take a well-trained LLM as the backbone, and thus this survey will not cover the details of pretraining language models. Towards the inference stage, as the input and output labels are all represented in interpretable natural language templates, there are multiple directions for improving ICL performance. This paper will give a detailed description and comparison, such as selecting suitable examples for demonstrations and designing specific scoring methods for different tasks.

We organize the current progress in ICL following the taxonomy above (as shown in Figure 2). With a formal definition of ICL (§3), we provide a detailed discussion of the warmup approaches (§4), the demonstration designing strategies (§5), and the main scoring functions(§6). §7 provides in-depth discussions of current explorations on unveiling the secrets behind the ICL. We further provide useful evaluation and resources for ICL (§8) and introduce potential application scenarios where ICL shows its effectiveness (§9). Finally, we summarize the challenges and potential directions (§10) and hope this could pave the way for researchers in this field.

3 Definition and Formulation

Following the paper of GPT-3 (Brown et al., 2020), we provide a definition of in-context learning: *In-context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstration*. Essentially, it estimates the likelihood of the potential answer conditioned on the demonstration by using a well-trained language model.

Formally, given a query input text x and a set of candidate answers $Y = \{y_1, \ldots, y_m\}$ (Y could be class labels or a set of free text phrases), a pretrained language model $\mathcal{M}$ takes the candidate answer with the maximum score as the prediction conditioning a demonstration set C. C contains an optional task instruction I and k demonstration examples; therefore, $C = \{I, s(x_1, y_1), \ldots, s(x_k, y_k)\}$ or $C = \{s(x_1, y_1), \ldots, s(x_k, y_k)\}$, where $s(x_k, y_k, I)$ is an example written in natural language texts according to the task. The likelihood of a candidate answer y_j could be represented by a scoring function f of the whole input sequence with the model $\mathcal{M}$:

$$P(y_j \mid x) \triangleq f_{\mathcal{M}}(y_j, C, x) \tag{1}$$

The final predicted label $\hat{y}$ is the candidate answer with the highest probability:

$$\hat{y} = \arg\max_{y_j \in Y} P(y_j|x). \tag{2}$$

The scoring function f estimates how possible the current answer is given the demonstration and the query text. For example, we could predict the class label in a binary sentiment classification by comparing the token probability of *Negative* and *Positive*. There are many f variants for different applications, which will be elaborated in §6.

According to the definition, we can see the difference between ICL and other related concepts. (1) Prompt Learning: Prompts can be discrete templates or soft parameters that encourage the model to predict the desired output. Strictly speaking, ICL can be regarded as a subclass of prompt tuning where the demonstration is part of the prompt. Liu et al. (2021) made a thorough survey on prompt learning. However, ICL is not included. (2) Few-shot Learning: few-shot learning is a general machine learning approach that uses parameter adaptation to learn the best model parameters for the task with a limited number of supervised examples (Wang and Yao, 2019). In contrast, ICL does not require parameter updates and is directly performed on pretrained LLMs.

4 Model Warmup

Although LLMs have shown promising ICL capability, many studies also show that the ICL capability can be further improved through a continual training stage between pretraining and ICL inference, which we call model warmup for short. Warmup is an optional procedure for ICL, which adjusts LLMs before ICL inference, including modifying the parameters of the LLMs or adding additional parameters. Unlike finetuning, warmup does not aim to train the LLM for specific tasks but enhances the overall ICL capability of the model.

4.1 Supervised In-context Training

To enhance ICL capability, researchers proposed a series of supervised in-context finetuning strategies by constructing in-context training data and multitask training. Since the pretraining objectives are not optimized for in-context learning (Chen et al., 2022a), Min et al. (2022b) proposed a method MetaICL to eliminate the gap between pretraining and downstream ICL usage. The pretrained LLM is continually trained on a broad range of tasks with demonstration examples, which boosts its few-shot abilities, e.g., MetaICL obtains performance comparable to supervised finetuning on 52 unique datasets.

Besides, recent work indicates the potential value of instructions (Mishra et al., 2021) and there is a research direction focusing on supervised instruction tuning. Instruction tuning enhances the ICL ability of LLMs through training on task instructions. Tuning the 137B LaMDA-PT (Thoppilan et al., 2022) on over 60 NLP datasets verbalized via natural language instruction templates, FLAN (Wei et al., 2022a) improves both the zero-shot and the few-shot ICL performance. Compared to MetaICL, which constructs several demonstration examples for each task, instruction tuning mainly considers an explanation of the task and is more easier to scale up. Chung et al. (2022) and Wang et al. (2022c) proposed to scale up instruction tuning with more than 1000+ task instructions.

4.2 Self-supervised In-context Training

To utilize raw corpus for warmup, Chen et al. (2022a) proposed to construct self-supervised training data according to the ICL formats in the downstream tasks. They transferred the original raw text into input-output pairs and considered four self-supervised objectives, including masked token prediction and classification tasks.

◇ **Takeaway**: *(1) Supervised training and self-supervised training both propose to train the LLMs before ICL inference. The key idea is to bridge the gap between pretraining and downstream ICL formats by introducing objectives close to in-context* *learning. Compared to in-context finetuning involving demonstration, instruction finetuning without a few examples as demonstration is simpler and more popular. (2) To some extent, these methods all improve the ICL capability by updating the model parameters, which implies that the ICL capability of the original LLMs has great potential for improvement. Therefore, although ICL does not strictly require model warmup, we recommend adding a warmup stage before ICL inference. (3) The performance advancement made by warmup encounters a plateau when increasingly scaling up the training data. This phenomenon appears both in supervised in-context training and self-supervised in-context training, indicating that LLMs only need a small amount of data to adapt to learn from the context during warmup.*

5 Demonstration Designing

Many studies have shown that the performance of ICL strongly relies on the demonstration surface, including demonstration format, the order of demonstration examples, and so on (Zhao et al., 2021; Lu et al., 2022). As demonstrations play a vital role in ICL, in this section, we survey demonstration designing strategies and classify them into two groups: demonstration organization and demonstration formatting, as shown in Table 1.

5.1 Demonstration Organization

Given a pool of training examples, demonstration organization focuses on how to select a subset of examples and the order of the selected examples.

5.1.1 Demonstration Selection

Demonstrations selection aims to answer a fundamental question: Which examples are good examples for ICL? We classify related studies into two categories, including unsupervised methods based on pre-defined metrics and supervised methods.

Unsupervised Method Liu et al. (2022) showed that selecting the closest neighbors as the in-context examples is a good solution. The distance metrics are pre-defined L2 distance or cosine-similarity distance based on sentence embeddings. They proposed KATE, a kNN-based unsupervised retriever for selecting in-context examples. In addition to distance metrics, mutual information is also a valuable selection metric (Sorensen et al., 2022). The advantage of mutual information is that it does not require labeled examples and specific LLMs. In addition, Gonen et al. (2022) attempted to choose prompts with low perplexity. Wu et al. (2022) selected the best subset permutation of kNN examples based on the code-length for data transmission to compress label y given x and C. This self-adaptive ranking method considered both selection and ordering. Levy et al. (2022) consider the diversity of demonstrations to improve compositional generalization. They select diverse demonstrations to cover different kinds of training demonstrations. Different from these studies selecting examples from human-labeled data, Kim et al. (2022a) proposed to generate demonstrations from LLM itself.

Supervised Method Rubin et al. (2022) proposed a two-stage retrieval method to select demonstrations. For a specific input, it first built an unsupervised retriever (e.g., BM25) to recall similar examples as candidates and then built a supervised retriever EPR to select demonstrations from candidates. A scoring LM is used to evaluate the concatenation of each candidate example and the input. Candidates with high scores are labeled as positive examples, and candidates with low scores are hard negative examples. Based on prompt tuning, Wang et al. (2023b) view LLMs as topic models that can infer concepts θ from few demonstrations and generate tokens based on concept variables θ. They use task-related concept tokens to represent latent concepts. Concept tokens are learned to maximize $P(y|x, \theta)$. They select demonstrations that are most likely to infer the concept variable based on $P(\theta|x, y)$. Besides, reinforcement learning was introduced by Zhang et al. (2022a) for example selection. They formulated demonstration selection as a Markov decision process (Bellman, 1957) and selected demonstrations via Q-learning. The action is choosing an example, and the reward is defined as the accuracy of a labeled validation set.

5.1.2 Demonstration Ordering

Ordering the selected demonstration examples is also an important aspect of demonstration organization. Lu et al. (2022) have proven that order sensitivity is a common problem and always exists for various models. To handle this problem, previous studies have proposed several training-free methods to sort examples in the demonstration. Liu et al. (2022) sorted examples decently by their distances to the input, so the rightmost demonstration is the closest example. Lu et al. (2022) defined the global and local entropy metrics. They found

Category	Methods	Demonstration Acquisition	LLMs	Main Tasks
Demonstration Selection	KATE (Liu et al., 2022)	Human design	GPT-3	SST, table-to-text
	SG-ICL (Kim et al., 2022a)	LM generated	GPT-J	SST, NLI
	EPR (Rubin et al., 2022)	Human design	GPT-{J, 3}/CodeX	Semantic parsing
Demonstration Ordering	GlobalE & LocalE (Lu et al., 2022)	Human design	GPT-{2, 3}	Text classification
Instruction Formatting	Self Instruct (Wang et al., 2022b)	LM generated	GPT-3/InstructGPT	SuperNaturalInstruction
Reasoning Steps Formatting	CoT (Wei et al., 2022c)	Human design	GPT-3/CodeX	Reasoning tasks
	AutoCoT (Zhang et al., 2022b)	LM generated	GPT-3/PaLM	Reasoning tasks
	Self-Ask (Press et al., 2022)	LM generated	GPT-3/InstructGPT	MultihopQA

Table 1: Summary of representative demonstration designing methods.

a positive correlation between the entropy metric and the ICL performance. With the entropy metric, they directly used the entropy metric to select the best ordering of examples.

5.2 Demonstration Formatting

A common way to format demonstrations is concatenating examples $(x_1, y_1), \ldots, (x_k, y_k)$ with a template $\mathcal{T}$ directly. However, in some tasks that need complex reasoning (e.g., math word problems, commonsense reasoning), it is not easy to learn the mapping from x_i to y_i with only k demonstrations. Although template engineering has been studied in prompting (Liu et al., 2021), some researchers aim to design a better format of demonstrations for ICL by describing tasks with the instruction I (§5.2.1) and adding intermediate reasoning steps between x_i and y_i (§5.2.2).

5.2.1 Instruction Formatting

Except for the well-designed demonstration examples, good instructions which describe the task precisely are also helpful to the inference performance. However, unlike the demonstration examples, which are common in traditional datasets, the task instructions depend heavily on human-written sentences. Honovich et al. (2022) found that given several demonstration examples, LLMs can generate the task instruction. According to the generation ability of LLMs, Zhou et al. (2022b) proposed Automatic Prompt Engineer (APE) for automatic instruction generation and selection. To further improve the quality of the automatically generated instructions, Wang et al. (2022b) proposed to use LLMs to bootstrap off its own generations. Existing work has achieved good results in automatically generating instructions, which provided opportunities for future research on combining human feedback with automatic instruction generation.

5.2.2 Reasoning Steps Formatting

Wei et al. (2022c) added intermediate reasoning steps between inputs and outputs to construct demonstrations, which are called chain-of-thoughts (CoT). With CoT, LLMs predict the reasoning steps and the final answer. CoT prompting can learn complex reasoning by decomposing input-output mappings into many intermediate steps. There are many pieces of research on CoT prompting strategies (Qiao et al., 2022) including prompt designing and process optimization. In this paper, we mainly focus on CoT designing strategies.

Similar to demonstration selection, CoT designing also considers CoT selection. Different from Wei et al. (2022c) manually writing CoTs, Auto-CoT (Zhang et al., 2022b) used LLMs with *Let's think step by step* to generate CoTs. In addition, Fu et al. (2022) proposed a complexity-based demonstration selection method. They selected demonstrations with more reasoning steps for CoT prompting.

As input-output mappings are decomposed into step-by-step reasoning, some researchers apply multi-stage ICL for CoT prompting and design CoT demonstrations for each step. Multi-stage ICL queries LLMs with different demonstrations in each reasoning step. Self-Ask (Press et al., 2022) allows LLMs to generate follow-up questions for the input and ask themselves these questions. Then the questions and intermediate answers will be added to CoTs. iCAP (Wang et al., 2022a) proposes a context-aware prompter that can dynamically adjust contexts for each reasoning step. Least-to-Most Prompting (Zhou et al., 2022a) is a two-stage ICL including question reduction and subquestion solution. The first stage decomposes a complex question into subquestions; in the second stage, LLMs answer subquestions sequentially, and previously answered questions and generated answers will be added into the context.

Scoring Function	Target	Efficiency	Task Coverage	Stability
Direct	$\mathcal{M}(y_j \mid C, x)$	+++	+	+
PPL	$\text{PPL}(S_j)$	+	+++	+
Channel	$\mathcal{M}(x \mid C, y_j)$	+	+	++

Table 2: Summary of different scoring functions.

◇ **Takeaway**: *(1) Demonstration selection strategies improve the ICL performance, but most of them are instance level. Since ICL is mainly evaluated under few-shot settings, the corpus-level selection strategy is more important yet underexplored. (3) For k demonstrations, the size of search space of permutations is $k!$. How to find the best orders efficiently or how to approximate the optimal ranking better is also a challenging question. (4) Adding chain-of-thoughts can effectively decompose complex reasoning tasks into intermediate reasoning steps. During inference, multi-stage demonstration designing strategies are applied to generate CoTs better. How to improve the CoT prompting ability of LLMs is also worth exploring (5) In addition to human-written demonstrations, the generative nature of LLMs can be utilized in demonstration designing. LLMs can generate instructions, demonstrations, probing sets, chain-of-thoughts, and so on. By using LLM-generated demonstrations, ICL can largely get rid of human efforts on writing templates.*

6 Scoring Function

The scoring function decides how we can transform the predictions of a language model into an estimation of the likelihood of a specific answer. A direct estimation method (Direct) adopts the conditional probability of candidate answers that can be represented by tokens in the vocabulary of language models (Brown et al., 2020). The answer with a higher probability is selected as the final answer. However, this method poses some restrictions on the template design, e.g., the answer tokens should be placed at the end of input sequences. Perplexity (PPL) is another commonly-used metric, which computes the sentence perplexity of the whole input sequence $S_j = \{C, s(x, y_j, I)\}$ consists of the tokens of demonstration examples C, input query x and candidate label y_j. As PPL evaluates the probability of the whole sentence, it removes the limitations of token positions but requires extra computation time. Note that in generation tasks such as machine translation, ICL predicts the answer by decoding tokens with the highest sentence probability combined with diversity-promoting strategies such as beam search or Top-p and Top-k (Holtzman et al., 2020) sampling algorithms.

Different from previous methods, which estimate the probability of the label given the input context, Min et al. (2022a) proposed to utilize channel models (Channel) to compute the conditional probability in a reversed direction, i.e., estimating the likelihood of input query given the label. In this way, language models are required to generate every token in the input, which could boost the performance under imbalanced training data regimes. We summarize all three scoring functions in Table 2. As ICL is sensitive to the demonstration (see §5 for more details), normalizing the obtained score by subtracting a model-dependent prior with empty inputs is also effective for improving the stability and overall performance (Zhao et al., 2021).

◇ **Takeaway**: *(1) We conclude the characteristics of three widely-used scoring functions in Table 2. Although directly adopting the conditional probability of candidate answers is efficient, this method still poses some restrictions on the template design. Perplexity is also a simple and widely scoring function. This method has universal applications, including both classification tasks and generation tasks. However, both methods are still sensitive to demonstration surface, while Channel is a remedy that especially works under imbalanced data regimes. (2) Existing scoring functions all compute a score straightforwardly from the conditional probability of LLMs. There is limited research on calibrating the bias or mitigating the sensitivity via scoring strategies. For instance, some studies add additional calibration parameters to adjust the model predictions (Zhao et al., 2021).*

7 Analysis

To understand ICL, many analytical studies attempt to investigate what factors may influence the performance and aim to figure out why ICL works. We summarize the factors that have a relatively strong correlation to ICL performance in Table 3 for easy reference.

7.1 What Influences ICL Performance

Pre-training Stage We first introduce influence factors in the LLM pretraining stage. Shin et al. (2022a) investigated the influence of the pretraining corpora. They found that the domain source is

Stage	Factor
Pretraining	Pretraining corpus domain (Shin et al., 2022a) Pretraining corpus combination (Shin et al., 2022a) Number of model parameters (Wei et al., 2022b; Brown et al., 2020) Number of pretraining steps (Wei et al., 2022b)
Inference	Label space exposure (Min et al., 2022c) Demonstration input distribution (Min et al., 2022c) Format of input-label pairing (Min et al., 2022c) Demonstration input-label mapping (Min et al., 2022c; Kim et al., 2022b) Demonstration sample ordering (Lu et al., 2022) Demonstration-query similarity (Liu et al., 2022)

Table 3: Summary of factors that have a relatively strong correlation to ICL performance.

more important than the corpus size. Putting multiple corpora together may give rise to emergent ICL ability, pretraining on corpora related to the downstream tasks does not always improve the ICL performance, and models with lower perplexity do not always perform better in the ICL scenarios. Wei et al. (2022b) investigated the emergent abilities of many large-scale models on multiple tasks. They suggested that a pretrained model suddenly acquires some emergent ICL abilities when it achieves a large scale of pretraining steps or model parameters. Brown et al. (2020) also showed that the ICL ability grows as the parameters of LLMs increase from 0.1 billion to 175 billion.

Inference Stage In the inference stage, the properties of the demonstration samples also influence the ICL performance. Min et al. (2022c) investigated that the influence of demonstration samples comes from four aspects: the input-label pairing format, the label space, the input distribution, and the input-label mapping. They prove that all of the input-label pairing formats, the exposure of label space, and the input distribution contribute substantially to the ICL performance. Counter-intuitively, the input-label mapping matters little to ICL. In terms of the effect of input-label mapping, Kim et al. (2022b) drew an opposite conclusion that correct input-label mapping does impact the ICL performance, depending on specific experimental settings. Lu et al. (2022) indicated that the demonstration sample order is also an important factor. In addition, Liu et al. (2022) found that the demonstration samples that have closer embeddings to the query samples usually bring better performance than those with farther embeddings.

7.2 Understanding Why ICL Works

Distribution of Training Data Concentrating on the pretraining data, Chan et al. (2022) showed that the ICL ability is driven by data distributional properties. They found that the ICL ability emerges when the training data have examples appearing in clusters and have enough rare classes. Xie et al. (2022) explained ICL as implicit Bayesian inference and constructed a synthetic dataset to prove that the ICL ability emerges when the pretraining distribution follows a mixture of hidden Markov models.

Learning Mechanism By learning linear functions, Garg et al. (2022) proved that Transformers could encode effective learning algorithms to learn unseen linear functions according to the demonstration samples. They also found that the learning algorithm encoded in an ICL model can achieve a comparable error to that from a least squares estimator. Li et al. (2023) abstracted ICL as an algorithm learning problem and showed that Transformers can implement a proper function class through implicit empirical risk minimization for the demonstrations. Other work attempted to build connections between ICL and finetuning. Taking linear regression as a starting point, Akyürek et al. (2022) found that Transformer-based in-context learners can implement standard finetuning algorithms implicitly. Also based on simple regression tasks, von Oswald et al. (2022) showed that linear attention-only Transformers with hand-constructed parameters and models learned by gradient descent are highly related. Dai et al. (2022) figured out a dual form between Transformer attention and gradient descent and further proposed to understand ICL as implicit finetuning. They compared GPT-based ICL and explicit finetuning on real tasks and found that ICL behaves similarly to finetuning from multiple perspectives.

Functional Modules Focusing on specific functional modules, Olsson et al. (2022) found that there exist some induction heads in Transformers that copy previous patterns to complete the next token. Further, they expanded the function of induction heads to more abstract pattern matching and completion, which may implement ICL. They provided several pieces of evidence on models with different sizes, which support their viewpoint that induction heads constitute the mechanism of ICL.

◇ **Takeaway**: *(1) Knowing and considering how ICL works can help us improve the ICL performance, and the factors that strongly correlate to ICL performance are listed in Table 3. (2) Although some analytical studies have taken a preliminary step to explain ICL, most of them are limited to simple tasks and small models. Extending analysis on extensive tasks and large models may be the next step to be considered. In addition, among existing work, understanding ICL as a process of meta-optimization seems to be a reasonable and promising direction for future research. If we build clear connections between ICL and meta-optimization, we can borrow ideas from the history of finetuning and optimization to improve ICL.*

8 Evaluation and Resources

8.1 Traditional Tasks

As a general learning paradigm, ICL can be examined on various traditional datasets and benchmarks, e.g., SuperGLUE (Wang et al., 2019), SQuAD (Rajpurkar et al., 2018). Implementing ICL with 32 randomly sampled examples on SuperGLUE, Brown et al. (2020) found that GPT-3 can achieve results comparable to state-of-the-art (SOTA) finetuning performance on COPA and ReCoRD, but still falls behind finetuning on most NLU tasks. Hao et al. (2022b) showed the potential of scaling up the number of demonstration examples. However, the improvement brought by scaling is very limited. At present, compared to finetuning, there still remains some room for ICL to reach on traditional NLP tasks.

8.2 New Challenging Tasks

In the era of large language models with in-context learning capabilities, researchers are more interested in evaluating the intrinsic capabilities of large language models without downstream task finetuning (Bommasani et al., 2021).

To explore the capability limitations of LLM on various tasks, Srivastava et al. (2022) proposed the BIG-Bench (Srivastava et al., 2022), a large benchmark covering a large range of tasks, including linguistics, chemistry, biology, social behavior, and beyond. The best models have already outperformed the average reported human-rater results on 65% of the BIG-Bench tasks through ICL (Suzgun et al., 2022). To further explore tasks actually unsolvable by current language models, Suzgun et al. (2022) proposed a more challenging ICL benchmark, BIG-Bench Hard (BBH). BBH includes 23 unsolved tasks, constructed by selecting challenging tasks where the state-of-art model performances are far below the human performances. Besides, researchers are searching for inverse scaling tasks,[1] that is, tasks where model performance reduces when scaling up the model size. Such tasks also highlight potential issues with the current paradigm of ICL. To further probe the model generalization ability, Iyer et al. (2022) proposed OPT-IML Bench, consisting of 2000 NLP tasks from 8 existing benchmarks, especially benchmark for ICL on held-out categories.

Benchmark	Tasks	#Tasks
BIG-Bench (Srivastava et al., 2022)	Mixed tasks	204
BBH (Suzgun et al., 2022)	Unsolved problems	23
PRONTOQA (Saparov and He, 2022)	Question answering	1
MGSM (Shi et al., 2022)	Math problems	1
LLMAS (Valmeekam et al., 2022)	Plan and reasoning tasks	8
OPT-IML Bench (Iyer et al., 2022)	Mixed tasks	2000

Table 4: New challenging evaluation benchmarks for ICL. For short, we use LLMAS to represent LLM Assessment Suite (Valmeekam et al., 2022).

Specifically, a series of studies focus on exploring the reasoning ability of ICL. Saparov and He (2022) generated an example from a synthetic world model represented in first-order logic and parsed the ICL generations into symbolic proofs for formal analysis. They found that LLMs can make correct individual deduction steps via ICL. Shi et al. (2022) constructed the MGSM benchmark to evaluate the chain-of-thought reasoning abilities of LLMs in multilingual settings, finding that LLMs manifest complex reasoning across multiple languages. To further probe more sophisticated planning and reasoning abilities of LLMs, Valmeekam et al. (2022) provided multiple test cases for evaluating various reasoning abilities on actions and change, where existing ICL methods

[1]https://github.com/inverse-scaling/prize

on LLMs show poor performance.

◇ **Takeaway**: *(1) Due to the restrictions of ICL on the number of demonstration examples, the traditional evaluation tasks must be adapted to few-shot settings; otherwise, the traditional benchmarks cannot evaluate the ICL capability of LLMs directly. (2) As ICL is a new paradigm that is different from traditional learning paradigms in many aspects, the evaluation of ICL presents new challenges and opportunities. Toward the challenges, the results of existing evaluation methods are unstable, especially sensitive to the demonstration examples and the instructions. Chen et al. (2022b) observed that existing evaluations by accuracy underestimate the sensitivity towards instruction perturbation of ICL. It is still an open question to conduct consistent ICL evaluation. Toward the opportunities for evaluation, as ICL only requires a few instances for the demonstration, it lowers the cost of evaluation data construction.*

9 Application

ICL manifests excellent performance on traditional NLP tasks and methods (Kim et al., 2022a; Min et al., 2022b). Especially, through demonstrations that explicitly guide the process of reasoning, ICL manifests remarkable effects on tasks that require complexity reasoning (Wei et al., 2022c) and compositional generalization (Zhou et al., 2022a). Unlike traditional optimization-based meta-learning approaches, Chen et al. (2022d) applied ICL for meta-learning. Their method adapts to new tasks through ICL, where model parameters are frozen, mitigating the challenging nested optimization problem.

Model Editing Apart from that, ICL has shown effectiveness for lightweight model editing. Si et al. (2022) analyzed whether it is possible to edit or adapt memorized knowledge in LLMs to new information through in-context demonstrations. They found that a large model scale and a mixture of all types of demonstration examples strengthen the knowledge editing success rate of ICL.

Data Annotation What's more, ICL has manifested the potential to be widely applied in data engineering. Benefiting from the strong ICL ability, it costs 50% to 96% less to use labels from GPT-3 than using labels from humans for data annotation. Combining pseudo labels from GPT-3 with human labels leads to even better performance at a small cost (Wang et al., 2021b).

Multimodality Due to the great success of ICL in the area of NLP, increasing literature works on adapting or applying the ICL ability to multimodal tasks, including text-to-speech synthesis (TTS) and visual question answering (Wang et al., 2023a; Hao et al., 2022a). For example, Wang et al. (2023a) treated text-to-speech synthesis as a language model task. They use audio codec codes as an intermediate representation and propose the first TTS framework with strong in-context learning capability. In addition to speech-language tasks, ICL can also be applied to vision-language tasks. Hao et al. (2022a) present METALM, a general-purpose interface to models across tasks and modalities. With a semi-causal language modeling objective, METALM is pretrained and exhibits strong ICL performance across various vision-language tasks.

Linear Probing Apart from ICL, linear probing is another way of black-box tuning (Sun et al., 2022) for LLMs, which learns a linear classifier based on final representations of LLMs and is suitable for full-data setting. Cho et al. (2022) propose prompt-augmented linear probing, a hybrid of linear probing and ICL. They train a linear classifier based on representations enhanced with prepend additional demonstrations. The hybrid of ICL and linear probing can cover the weakness of each other and scale the ICL to the full-data setting.

10 Challenges and Future Directions

10.1 New Pretraining Strategies

As investigated by Shin et al. (2022b), language model objectives are not equal to ICL abilities. Researchers have proposed to bridge the gap between pretraining objectives and ICL through intermediate tuning before inference (Section 4), which shows promising performance improvements. To take it further, tailored pretraining objectives and metrics for ICL have the potential to raise LLMs with superior ICl capabilities.

10.2 Distill the ICL Ability to Smaller Models

Previous studies have shown that in-context learning for reasoning tasks emerges as the scale of computation and parameter exceed a certain threshold (Wei et al., 2022b). Transferring the ICL ability to smaller models could facilitate the model deployment greatly. Magister et al. (2022) showed that it is possible to distill the reasoning ability to small language models such as T5-XXL. The distillation is achieved by finetuning the small model on the chain-of-thought data (Wei et al., 2022c) generated by a large teacher model. Although promising performance is achieved, the improvements are likely task-dependent. Further investigation on improving the reasoning ability by learning from larger LLMs could be an interesting direction.

10.3 Knowledge Augmentation and Updating

ICL presents new issues for enhancing and updating knowledge in LLMs.

Knowledge Augmentation The knowledge of LLMs is entirely derived from the pretrained corpus. Therefore, LLMs may lack certain knowledge and generate hallucinations during ICL inference, especially towards long-tailed factual knowledge or commonsense knowledge rarely described in texts. Therefore, it is essential to augment knowledge for ICL. Different from traditional work, which adds knowledge adapters (Wang et al., 2021a) or provides structured knowledge during pretraining (Zhang et al., 2019; Peters et al., 2019), retrieving correct knowledge and integrating the correct knowledge with the context in a lightweight manner is possibly promising for ICL.

Knowledge Updating Although LLMs can serve as knowledge bases, the knowledge in LLMs could be wrong or out-of-date. However, it is costly to retrain the LLMs with up-to-date data. Si et al. (2022) made an initial trial on in-context knowledge updating. By providing counterfactual examples in the demonstration, they found that GPT-3 updates its answers around 85% of the time and larger models are better at in-context knowledge updating. However, this approach may affect other correct knowledge in LLMs. Compared with knowledge editing for finetuned models (De Cao et al., 2021), the impact of in-context knowledge updating on paraphrased facts and irrelevant facts has not been rigorously evaluated. Updating the wrong or out-of-date knowledge for ICL is worth further exploration.

10.4 Robustness to Demonstration

Previous studies have shown that ICL performance is extremely unstable, from random guess to SOTA, and can be sensitive to many factors, including demonstration permutation, demonstration format, etc. (Zhao et al., 2021; Lu et al., 2022). The robustness of ICL is a critical yet challenging problem.

However, most of the existing methods fall into the dilemma of accuracy and robustness (Chen et al., 2022c), or even at the cost of sacrificing inference efficiency. To effectively improve the robustness of ICL, we need deeper analysis of the working mechanism of the ICL. We believe that the analysis of the robustness of the ICL from a more theoretical perspective rather than an empirical perspective can highlight future research on more robust ICL.

10.5 ICL for Data Engineering

Compared to human annotation (e.g., crowd-sourcing) or noisy automatic annotation (e.g., distant supervision), ICL generates relatively high quality data at a low cost (mentioned in Section 9). First, ICL takes only a few examples to learn the data engineering objective, which saves the cost of annotating training data. Second, the strong reasoning ability and text-generating ability of LLM show great potential to generate high-quality data.

However, how to use ICL for data annotation remains an open question. For example, Ding et al. (2022) performed a comprehensive analysis and found that generation-based methods are more cost-effective in using GPT-3 than annotating unlabeled data via ICL. We believe that improving ICL for data annotation is a direction with practical value, and ICL will be a new paradigm in data annotation, data augmentation, data pruning, as well as adversarial data generation.

11 Conclusion

In this paper, we survey the existing ICL literature and provide an extensive review of advanced ICL techniques, including training strategies, demonstration designing strategies, evaluation datasets and resources, as well as related analytical studies. Furthermore, we highlight critical challenges and potential directions for future research. To the best of our knowledge, this is the first survey about ICL. We hope this survey can highlight the current research status of ICL and shed light on future work on this promising paradigm.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022.

What learning algorithm is in-context learning? investigations with linear models. *CoRR*, abs/2211.15661.

Richard Bellman. 1957. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *CoRR*, abs/2205.05055.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022a. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3558–3573, Seattle, United States. Association for Computational Linguistics.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2022b. On the relation between sensitivity and accuracy in in-context learning. *ArXiv preprint*, abs/2209.07661.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen R. McKeown, and He He. 2022c. On the relation between sensitivity and accuracy in in-context learning. *CoRR*, abs/2209.07661.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022d. Meta-learning via language model in-context tuning. In *Proc. of ACL*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. *arXiv preprint arXiv:2212.10873*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. *CoRR*, abs/2212.10559v2.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proc. of EMNLP*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Linlin Liu, Lidong Bing, Shafiq Joty, and Boyang Li. 2022. Is gpt-3 a good data annotator?

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

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? A case study of simple function classes. *CoRR*, abs/2208.01066.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *ArXiv preprint*, abs/2212.04037.

Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. 2022a. Language models are general-purpose interfaces. *arXiv preprint arXiv:2206.06336*.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022b. Structured prompting: Scaling in-context learning to 1,000 examples. *ArXiv preprint*, abs/2212.06713.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proc. of ICLR*. OpenReview.net.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2022. Instruction induction: From few examples to natural language task descriptions. *CoRR*, abs/2205.10782.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2022a. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *ArXiv preprint*, abs/2206.08082.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022b. Ground-truth labels matter: A deeper look into input-label demonstrations. *CoRR*, abs/2205.12685.

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

Yingcong Li, M. Emrullah Ildiz, Dimitris S. Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *CoRR*, abs/2301.07067.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. 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*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. of ACL*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *ArXiv preprint*, abs/2212.08410.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proc. of ACL*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. 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*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *CoRR*, abs/2209.11895.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. *ArXiv preprint*, abs/2212.07677.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proc. of EMNLP*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *CoRR*, abs/2210.03350.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. of ACL*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. 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*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *ArXiv preprint*, abs/2210.01240.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *ArXiv preprint*, abs/2210.03057.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. 2022a. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5168–5186, Seattle, United States. Association for Computational Linguistics.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. 2022b. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5168–5186, Seattle, United States. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *ArXiv preprint*, abs/2210.09150.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proc. of ACL*, pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. *ArXiv preprint*, abs/2201.03514.

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. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *ArXiv preprint*, abs/2210.09261.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *ArXiv preprint*, abs/2201.08239.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *ArXiv preprint*, abs/2206.10498.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Boshi Wang, Xiang Deng, and Huan Sun. 2022a. Iteratively prompt pre-trained language models for chain of thought. In *The 2022 Conference on Empirical Methods for Natural Language Processing*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021b. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yaqing Wang and Quanming Yao. 2019. Few-shot learning: A survey.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language model with self generated instructions. *ArXiv preprint*, abs/2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022c. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *Proc. of ICLR*. OpenReview.net.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent abilities of large language models. *CoRR*, abs/2206.07682.

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

Patrick H Winston. 1980. Learning and reasoning by analogy. *Communications of the ACM*, 23(12):689–703.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *Proc. of ICLR*. OpenReview.net.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022a. Active example selection for in-context learning. *CoRR*, abs/2211.04486.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proc. of ACL*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *CoRR*, abs/2210.03493.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022a. Least-to-most prompting enables complex reasoning in large language models. *ArXiv preprint*, abs/2205.10625.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. *ArXiv preprint*, abs/2211.01910.