Retrieval-Augmented In-Context Learning with Large Language Models

Aristides Milios, School of Computer Science

McGill University, Montreal

April 2024

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Computer Science

Abstract

One of the main unexpected abilities of large autoregressive language models (LLMs) was their ability to perform in-context learning (ICL). Through ICL, LLMs learn to perform tasks via task demonstrations directly in their input context, without any parameter updates being necessary. In this work, we seek to expand the types of tasks that ICL can be applied to, by augmenting LLMs with secondary dense retrieval models. These retrieval models allow us to use ICL for classification tasks whose specification cannot fit in the context window of the model, by retrieving a limited subset of demonstrations dynamically. We investigate various complex retrieval strategies, that aim to balance demonstration diversity, class representation, and other factors. By applying LLMs to short text classification problems with retriever models, we achieve SOTA performance across four datasets and two domains.

We use this same retrieval framework to investigate whether models can learn to follow instructions through ICL. We evaluate if an LLM can generalize zero-shot to a set of natural language tasks using only a small set of demonstration tasks when provided only a task description. This ability, commonly known as instruction following, usually requires supervised fine-tuning (SFT) or reinforcement learning from human feedback (RLHF); in this thesis, we examine if it is achievable through pure ICL only. We demonstrate that irrelevant task demonstrations provided in-context greatly boost the performance of base LLMs on held-out tasks, both through term-based and model-based evaluation. Regardless, a gap with RLHF-tuned models remains. However, we demonstrate that the prompting method is complimentary to RLHF and SFT-based tuning, and show the strongest results among all models by combining prompting and an RLHF-tuned model.

Abrégé

L'une des principales capacités inattendues des grands modèles de langage autorégressifs (LLM) était leur capacité à effectuer un apprentissage en contexte (ICL). Grâce à ICL, les LLM apprennent à effectuer des tâches via des démonstrations de tâches directement dans leur contexte d'entrée, sans qu'aucune mise à jour des paramètres ne soit nécessaire. Dans ce travail, nous cherchons à élargir les types de tâches auxquelles ICL peut être appliqué, en augmentant les LLM avec des modèles secondaires de récupération dense. Ces modèles de récupération nous permettent d'utiliser ICL pour des tâches de classification dont la spécification ne peut pas tenir dans la fenêtre contextuelle du modèle, en récupérant dynamiquement un sous-ensemble limité de démonstrations. Nous étudions diverses stratégies de récupération complexes, qui visent à équilibrer la diversité des démonstrations, la représentation des classes et d'autres facteurs. En appliquant les LLM à des problèmes de classification de textes courts avec des modèles de récupération, nous obtenons des performances SOTA sur quatre ensembles de données et deux domaines.

Nous utilisons ce même cadre de récupération pour déterminer si les modèles peuvent apprendre à suivre des instructions via ICL. Nous évaluons si un LLM peut généraliser le zero-shot à un ensemble de tâches en langage naturel en utilisant uniquement un petit ensemble de tâches de démonstration lorsqu'il est fourni uniquement une description de la tâche. Cette capacité, communément appelée suivi d'instructions, nécessite généralement un réglage fin supervisé (SFT) ou un apprentissage par renforcement à partir de la rétroaction humaine (RLHF) ; dans cette thèse, nous examinons si cela est réalisable uniquement grâce à l'ICL pure. Nous démontrons que les démonstrations de tâches non pertinentes fournies en contexte améliorent considérablement les performances des LLM de base sur les tâches suspendues, à la fois par le biais d'une évaluation

basée sur des termes et sur un modèle. Quoi qu'il en soit, il reste un écart avec les modèles optimisés pour le RLHF. Cependant, nous démontrons que la méthode d'invite est complémentaire du réglage basé sur RLHF et SFT, et montrons les résultats les plus forts parmi tous les modèles en combinant l'invite et un modèle optimisé par RLHF.

Previously Published Material

The material in Chapter 3 was published at the GenBench Workshop at EMNLP 2023 [29]. As sole first author, Aristides Milios performed all the work presented in this Chapter, including experimental design, development of methodology, evaluation and analysis of results.

Acknowledgements

I am grateful to my first supervisor Siva Reddy for supporting me throughout the duration of my Master's, providing expert guidance, but also never hesitating to give me opportunities to connect with other researchers and foster collaborations. I thank him in particular for giving me many opportunities to truly experience the research scene, travelling abroad and connecting with fellow researchers at other institutions around the world.

I am very grateful to my co-supervisor Dzmitry Bahdanau as well for supporting me for the last year and a half of my Master's. Dima's guidance and expertise were paramount to me becoming a fully-fledged researcher, and I am extremely thankful for the opportunities he provided me.

I thank my parents, Evangelos Milios and Evangelia Tastsoglou, for always supporting me no matter what, giving me the best advice they could offer and never hesitating to provide a helping hand when it was needed. I thank my father for being a constant standard to aspire to, pushing me to always strive to greater heights.

Thank you to all of my labmates at Siva's lab, you were all a pleasure to be around for two and a half years. I shared many great moments with all of you, especially at our lovely yearly retreats into the Quebec wilderness. These are memories I will cherish forever.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the advent of large pre-trained autoregressive language models (LLMs), a new mode of learning appeared on the scene: in-context learning (ICL), or learning without any parameter updates [4]. In ICL, a few demonstrations of input and output for a specific task the model has not been trained for are passed directly into the model's input context window. The advent of ICL set the stage to make NLP much more accessible to a wider audience than before. No longer would a custom model need to be meticulously tuned for a downstream task. With ICL, a layperson can write a few demonstrations and achieve reasonably good performance for their task. Beyond the factor of accessibility to the average tech-savvy individual, ICL greatly aided NLP practitioners as well, both because training is computationally expensive, but also because ICL unlocks the use of LLMs for low-resource tasks (where there is limited data—in other words, it allows for few-shot learning).

In-context learning also has several limitations. One limitation, intimately tied to the fact that the theoretical understanding of *why* ICL works is still somewhat lacking, is the brittleness of prompts and examples [3, 14, 27, 42]. Specifically, when we refer to "brittleness", what we mean is the unpredictability of the resulting performance based on the contents of the prompt. This brittleness can be in relation to which examples are provided in-context, as well as the ordering of the examples that are provided. Existing work has shown [27] that some permutations give SOTA performance

while others may be near random chance, for the same set of in-context examples. The result of this was the birth of the "prompt engineer": because prompts could not be trusted to "just work", a role sprang up to tune them to the highest possible performance. Another serious limitation of ICL is the size of the context window. Although models with longer and longer context windows are being trained constantly, the ability for a model to make effective use of a long context window is still elusive [24]. As such, ICL is typically limited to tasks whose entire "specification" can be held in the context window. In the case of text classification tasks with just two or a few labels, it is simple to fit several examples of each class in the prompt provided to the model, and thereby provide the model enough information to perform the task via ICL. However, in the case of a text classification task of several tens or even hundreds of labels, it becomes infeasible to fit enough examples of each class into the context window for the model to be able to perform the task.

The motivation for this thesis is to examine **how ICL functions in the context of a secondary retrieval model**, as well as to investigate **what kinds of tasks models can learn via ICL**. We know that LLMs revolutionized the field of NLP through their excellent performance on a wide gamut of NLP tasks. The benefits of incorporating a retrieval model are several:

1. It allows us to unlock LLM-level performance on tasks that it was previously difficult to apply LLMs to due to context length limitations,

2. It allows us to increase the computational efficiency of ICL, by providing fewer but more meaningful examples to the LLM

3. It provides a more flexible approach to ICL, allowing us to dynamically update the retrieval set and have the right examples be fetched by the retrieval model at test time based on the query

Along with ICL, the advent of large pre-trained language models eventually unlocked another mode of function: zero-shot instruction-based task execution. This was the ultimate goal: to have a system that can perform tasks with no task demonstrations at all, only through a natural language description of the task. This mode of function was an important step forward for NLP: zero-shot

task generalization can be seen as relatively close to human function, where one human can explain to another in words how to perform a task. Research towards this goal had been performed in the context of massively pre-trained encoder-decoder models (T5 [35], Flan-T5 [9]), however these models were limited in the amount of text they could generate due to scaling constraints at the time and their architecture, and as such were not able to be applied to long-form generation or dialogue easily. It was really the combination of this goal with large autoregressive language models for dialogue that unlocked the highest level of usability. The main paradigm that was settled on was massively pre-training autoregressive models (producing a "base model"), then performing "instruction tuning" on them, to allow them to shift from few-shot task execution to zero-shot execution based on descriptions (instructions). In addition to this, the dialogue-based interaction format was adopted widely, combining zero-shot task execution with an existing well-known chatbot UI paradigm.

LLMs in their default pre-training state are unable to demonstrate this capability solely from the next-word prediction, as the most probable continuations from their pre-training data are not necessarily performing the user's task as helpfully as possible (this is referred to in the literature as "misalignment" [33] between the objectives of being a helpful assistant, and best predicting the next word of the training corpus). As such, ordinary LLMs must be fine-tuned for instruction following, after which they can demonstrate the ability to follow instructions directly. However, such tuning is often very costly. There are two main paradigms, supervised fine-tuning and reinforcement learning from human feedback (RLHF)-based tuning. In the former, large amounts of instruction following data are necessary. In the latter, both instruction following data is necessary, as RLHF usually includes a supervised fine-tuning first step, and then also large amounts of human preference data is necessary, to tune the model on these preference and reach a model state that follows human expectation (the concept of "alignment").

In this work, we leverage the insights gleaned from the experiments on ICL with retrieval on large label set classification tasks, to see if we can perform in-context instruction following. Simultaneously, whether or not models can learn to instruction-follow in-context tells us something about how ICL works, specifically with regards to the "superficial alignment hypothesis" [56]. The

superficial alignment hypothesis states that alignment (and by extension instruction tuning) usually constitutes tweaking surface-level features of the model's output (e.g. style of response, etc.) and does not teach the model to perform any task it did not already acquire the ability to do during pre-training. If models are able to meaningfully learn instruction-following in-context (i.e. are able to zero-shot generalize to novel tasks only through a prompt, especially from only a few examples), this would be evidence in favor of this hypothesis. This ability to "learn how to learn" in-context would support the idea that fine-tuning, while useful for optimizing performance on specific tasks or adjusting the model's output style, is not strictly necessary for the model to understand and perform new tasks.

1.2 Statement of Contributions

In this work, we demonstrate that using retrieval models in conjunction with LLMs allows us to apply LLMs to text classification tasks with large label sets. By leveraging the pre-trained knowledge of LLMs through ICL, we reach SOTA performance across 4 different short text classification datasets, and either match or in most cases outperform fine-tuned adapter-based methods with the same amount of data. We perform detailed analysis over the number of demonstrations, showing that larger models can better take advantage of longer context sizes (more demonstrations). We perform ablations to investigate what parts of the input the model is using for ICL, demonstrating that everything plays a role, including the semantic similarity of the demonstrations to the query. We experiment with various selection methods that aim to balance diversity and class representation in the prompt, showing that pure nearest neighbor is consistently the most effective out of all approaches tested. We show that least-to-most-semantically-similar demonstration ordering is the most effective across all datasets.

In addition to the experiments on short text classification, we apply the same retrieval with ICL framework to the meta-task of instruction following, to see if models are able to learn to follow instructions zero-shot without any parameter updates. In these experiments, we provide the model with irrelevant task demonstrations in-context, and see if it can perform held-out unseen tasks with

nothing except a task description. We conclude that we are indeed able to significantly improve performance on the SuperNaturalInstructions multi-task benchmark. We perform model-based evaluation, showing that although the term-based metrics demonstrate stronger performance than an RLHF-tuned model, in truth there is still a gap between the RLHF-tuned model and prompted base models. However, we demonstrate that the RLHF-tuned model can be prompted with irrelevant task examples as well, and demonstrates significantly increased performance as a result, showing that prompting and ordinary RLHF-tuning are complimentary approaches.

1.3 Chapter Overview

In Chapter 2, we provide an overview of the related background literature to this work, contextualizing them in relation to the thesis. In Chapter 3, we present and discuss the results of applying retrieval with ICL to short text classification tasks. In Chapter 4, we present and discuss applying the same framework to the meta-task of instruction following. Finally, in Chapter 5, we tie these two chapters together, discussing what broader conclusions we can come to about ICL from the experiments in both chapters, and discuss potential future work to continue investigating ICL in the context of retrieval.

Chapter 2

Background

In this chapter, we provide the relevant context necessary to understand the work presented in this thesis. First we discuss few-shot in-context learning: what it is, and attempts to understand how exactly it functions. We talk about existing works on selecting demonstrations and ordering them most effectively in the prompt. We briefly go over the language models used in this thesis, then talk about retrieval broadly, giving some context to the use of neural networks in the space. We then go over the specific tasks tested in this thesis: intent detection, sentiment classification, and the meta-task of instruction following. Finally, we talk about the Superficial Alignment Hypothesis, and briefly go into the metrics for evaluating instruction-following.

2.1 Few-shot In-Context Learning

Few-shot in-context learning is a relatively recent advance in NLP research, enabled by massive large language models (LLMs). First introduced with the release of GPT-3, it was demonstrated that a series of examples could be provided directly in the input context of a Transformer language model, and the model would follow the pattern and provide a prediction for a novel query following the same format [5]. Since then, many attempts have been made to understand how exactly in-context learning (hereafter referred to as "ICL") works, through various investigations and ablations.

In [30], authors investigate the mechanisms behind ICL by perturbing the labels provided in the prompt. Specifically, they demonstrate that for a set of classification and multi-choice datasets, performance is not significantly affected when random class names are provided instead of the true ground-truth classes. The implication of this result is that LLMs are not really learning the input-output correspondence from the examples provided, but rather are learning more superficial elements from the examples provided in-prompt, for example the output space (set of classes) or the input distribution (the distribution of examples). Other works, for example [15], attempt to discover what classes of functions are learnable by LLMs in-context. This specific work demonstrates that LLMs are able to learn most simple function classes in-context fairly reliably, including under train-time-to-test-time distribution shifts. In [53], authors examine ICL with the framing of Bayesian inference, creating a synthetic dataset with long-range dependencies to fully examine under which conditions ICL emerges. They demonstrate that several phenomena arise with the synthetic dataset that are also present in the large real-world datasets that LLMs are trained on, such as order sensitivity, and better ICL correlated with scale. Another work [7] reaches a similar conclusion, providing evidence that certain properties must be present in the pre-training data to cause a model to exhibit ICL abilities. Specifically, they discover that the Zipf distribution and "burstiness" (entities appearing clustered over time) are key ingredients. They also discover that only Transformers seem to exhibit ICL, and not recurrent networks, and as such, both architecture and correct data are necessary to produce models capable of ICL.

In [38], authors provide evidence that ICL is not caused by robust reasoning abilities, but rather is directly correlated with the frequency of certain terms appearing in the pre-training dataset. Specifically, they demonstrate that performance on arithmetic tasks directly correlates with the frequency of the given numbers appearing in the pre-training dataset (in other words, numbers appearing less often in the training data yielded worse performance when used as operands in arithmetic operations). This implies that LLMs are not generalizing robustly beyond the data they are fed at training time.

2.1.1 Demonstration Selection Methods

One of the earliest studies of the role of example selection in ICL is "KATE" [23]. In this paper, the authors probe the performance of GPT-3 on NLP tasks using KNN retrieval (RoBERTa) for example selection. They compare this method against random selection and using the retrieval model directly (plain KNN). They also examine the effect of example ordering on performance and conclude that the most performant ordering (least-to-most and most-to-least similar orderings are tested) depends on the dataset. In this thesis, we also experiment with example ordering, and conclude that least-to-most ordering is the most effective across all datasets and tasks tested.

2.1.2 Works demonstrating ordering instability

Several recent works have demonstrated that the order of in-context examples makes a larger difference in performance, including [27, 55]. These works demonstrate such order instability that certain permutations bring near SoTA performance on tasks while others perform at near random guessing.

2.2 Language Models used in this Thesis

In this thesis, the main two language model families used are OPT [54] and LLaMA [47, 48]. OPT stands for "Open Pre-trained Transformer", and is the first major large language model whose weights were made widely accessible. OPT was developed by Meta AI. OPT was trained on a corpus of roughly 180B tokens, a combination of all the datasets from RoBERTa [26], the Pile, and Reddit data. It showed comparable performance to the original GPT-3.

In 2022, DeepMind releases the "Chinchilla" family of models [17]. The DeepMind authors claimed that LLMs at the time were undertrained, in the sense that with the number of parameters they have (already in the hundreds of billions at the time), they could be trained on much more data, such that they would see their loss continue to improve. The authors created a set of "compute-optimal" predictions, showing how to balance model size vs. training data for a given level of compute to get the optimally performant LLM.

Soon after, Meta AI released their second major open large language model: LLaMA. LLaMA took the lessons from Chinchilla and trained a family of models with far more data than before. Whereas OPT was trained on 180B tokens, LLaMA-1 was trained on 1 to 1.4 trillion tokens, depending on model size. This massive pre-training dataset was composed mostly of Common Crawl [1] (67%) but included a mixture of GithuB, Wikipedia, Books, ArXiv and StackExchange as well. The authors of LLaMA mention that the loss shows no signs of plateauing at 1 trillion tokens, suggesting that the dataset size could be made even larger. Sometime later, Meta AI releases the second version of the model family: LLaMA-2. With LLaMA-2, Meta AI once again increases the dataset size to 2 trillion tokens, training all the models with a 4K context length (instead of the 2K length of the original LLaMA). One of the major developments with LLaMA-2 was the release of "chat" versions of all the models. These chat versions have been tuned with reinforcement learning from human feedback (RLHF [34]), after a supervised fine-tuning stage. More details about instruction following are provided in Section 2.6.

In this work, we primarily use LLaMA-1 as well as the base (non-RLHF-tuned) version of LLaMA2. In the work presented in Chapter 4, we use the RLHF-tuned version of LLaMA-2 as a point of comparison against our ICL-based instruction following method.

2.3 Retrieval

Information retrieval (IR) is a field with a long and storied history completely independent of machine learning. IR as a field is primarily focused on information discovery from a set of documents. In this thesis, retrieval systems are used for short texts to retrieve examples as a way to enhance ICL-based predictive approaches. In this section, we briefly outline older methods for IR before diving into more recent ML-enabled advances.

[1] https://commoncrawl.org/

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

where:

$$\text{IDF}(t, D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right)$$

Figure 2.1: The TF-IDF equation. $\text{TF}(t, d)$ is the number of times term t appears in document f, divided by the length of the document. D is the entire corpus, N is the number of documents in the corpus, and $|\{d \in D : t \in d\}|$ is the number of documents t appears in.

2.3.1 Classical Term-Based Retrieval Methods

Before neural methods began to be used, IR primarily relied on term-based retrieval methods. In this case, term-based refers to the fact that these approaches rely on the direct character overlap (surface-form similarity) between specified search/query terms and the documents being searched through. Two of the most well-known of these methods are TF-IDF (the foundations of which were created in the 1970s [46]) and BM25 [43]. Both methods are used as the ranking function component of a complete IR system. TF-IDF is composed of two components (see Equation 2.1), the term frequency (TF) component, which measures how often a term appears in a document, and the inverse document frequency (IDF) component, which measures how "important" a given term is. Terms that appear often in many documents (e.g. articles, pronouns, etc.) are given a lower IDF. Finally, the TF-IDF score is the product of the two components. Terms that are rare over the full set of documents but appear frequently in a specific document score the highest. To calculate relevance to a query, documents are ranked by the sum of the TF-IDF scores of each of the terms in the query. BM25 works similarly to TF-IDF, and can be thought of as a more sophisticated version of it. In this thesis, BM25-based retrieval is used as a baseline in comparison with neural (dense) retrieval methods.

2.3.2 Dense Retrieval

Using neural networks over classical term-based retrieval approaches provides several major advantages. One of the main advantages relates to semantic understanding. As mentioned previously,

classical methods simply look at term overlap between documents and the query. If a user types "chicken", but all the documents refer to "poultry", what are likely the most relevant documents will not be returned as there is no mechanism by which to take semantic similarity into account. On the contrary, neural methods are able to look at the actual semantic similarity between terms, as measured by the closeness of their representations in the neural model's representation space. This allows them to take synonyms and polysemy (the same surface form yielding multiple meanings) into account in a way term-based methods cannot by definition. Beyond this, neural models take into account the context and order of words when producing document representations, while term-based methods simply treat documents as bags-of-words. Again, this more nuanced handling can better capture meaning and yield better retrieval results.

One of the first tasks BERT [11] was tested on was semantic textual similarity (STS), setting the stage for later neural retrieval works. One of the first major works to popularize neural retrieval as such was Sentence-BERT [39]. In this work, a siamese network structure is used to train and perform inference using BERT networks, enabling fast and accurate retrieval where the query and document are fed to the same retrieval model separately, producing two output representations which are then compared by a similarity function (e.g. cosine similarity). An alternative but similar scheme is employed by DPR [20] for the purposes of Question Answering (QA), where rather than training a single retrieval model, two separate encoders are trained, one for the query and one for the document, due to the asymmetric nature of the retrieval (i.e. queries and documents follow different distributions). In this thesis we primarily employ Sentence-BERT as our retrieval model, as our retrieval case is symmetric (we are looking for the similarity between a test-time query and a set of example queries, to retrieve the best possible demonstrations for ICL).

2.4 Intent Detection

Intent detection is one component of a task-oriented dialogue system (e.g. a virtual assistant), and a key task in the field of NLP. The goal of intent detection is to classify a user utterance (short text snippet) into one of many intent categories, i.e. to identify what exactly the task the user wishes to

perform is. Examples of tasks may be to make a calendar event, make a reservation at a restaurant, or pay a bill. Given the open-ended nature of task-oriented dialogue systems, the number of intent categories in most datasets for this task can be anywhere between 60 and 150. After correctly tagging a user utterance into one of these intent categories, a complete task-oriented dialogue system would then pass this data onto another system that either performs the associated action or generates clarifying questions and/or responses.

The current state of the art in few-shot intent detection is the ConvFit method [49]. ConvFit uses a pre-trained LM in a dual-encoder configuration (e.g. BERT or RoBERTa) with two training stages. The first stage is a conversational fine-tuning stage using a generic conversational corpus with a retrieval task (using tuples of (context, response) retrieve the correct response for each context). The second stage is fine-tuning on the specific intent classification dataset with a contrastive loss, allowing the resulting LM to be used in a KNN fashion.

Intent detection cannot easily be applied for use directly with autoregressive ICL-capable LLMs due to the limitation of context length. With so many classes, fitting more than a single demonstration per class quickly reaches the limit of the input length of these models. As such, existing works leveraging LLMs for intent detection typically use other approaches. In [44], LLMs are used for data augmentation of existing intent detection datasets. The authors demonstrate that performing data augmentation via LLM leads to the largest gains when the intent categories are more distinct. In cases of fine-grained similar intent categories, the LLM augmentation leads to smaller gains, which the authors hypothesize is due to the LLM generating utterances of similar classes instead of the one requested (in other words, the LLM has difficulty separating the fine-grained intent classes for the purposes of augmentation).

2.5 Sentiment Classification

Sentiment classification is a relatively old task in NLP that concerns identifying the sentiment expressed by an author in text. Most traditional older approaches relied on building sets of positively and negatively charged terms, then providing a prediction based on identifying which of these

terms were more present in a given piece of text. For a long time the task was framed exclusively as a binary prediction task, or at best, a 3-way classification task, with the addition of a "neutral" class to the mix. More recently, the problem has begun to be framed in a more nuanced way, including multiple emotion classes such as anger, fear, and happiness (e.g. using the Ekman emotion taxonomy [13]). In this work, we use the GoEmotions dataset [10], which uses a 28-way classification setup.

2.6 Instruction Following

Instruction-following on a high level refers to the ability of an LLM to follow instructions expressed directly in the prompt as directives (e.g., do X, write Y) in zero-shot (i.e. without a series of input/output examples). This is in contrast to the "default" (from pre-training) behavior of an LLM, which is simply to predict the most likely next token based on the context. When "base" (no additional tuning) LLMs are given direct instructions in a zero-shot way, very rarely will they be able to perform the task they are instructed to do, and are much more likely to simply output related questions, queries, or something else along these lines (pulling from pre-training context where lists of questions or instructions are provided together).

2.6.1 Training for Instruction Following

Given that base LLMs do not exhibit this ability, various ways exist to instill this ability into the model. Two frequently confused concepts are instruction following and the more general principle of "alignment". "Alignment" refers to the idea that the model should behave in ways its users intuitively expect it to, and should generally follow human values and moral guidelines. In this sense, instruction following is in effect a "subset" of alignment, in that an aligned model is able to follow prompt instructions zero-shot. However, models can be trained simply to follow instructions, and not necessarily be aligned in terms of moral values. Alignment has yet to have a widely-accepted rigorous definition in the literature. Existing works on alignment often tackle alignment in terms of optimizing against human preferences, without defining more specific criteria or desiderata

(e.g. reinforcement learning from human preferences or RLHF, [34]). In this thesis, we primarily approach the problem of instruction following specifically, and do not touch the wider issue of how to align models most effectively in terms of moral expression, refusal of immoral queries, etc.

One of the first papers to introduce the concept of tuning for instruction following is Natural Instructions [32]. In this work, the authors crowdsource 61 distinct NLP tasks to form a dataset that can be used to fine-tune LLMs. They show that fine-tuning a BART model along with human-written instructions for each task improves cross-task generalization for tasks unseen at training time. The same work is extended in [51], which is the version of the dataset used in Chapter 4 of this thesis. The extension greatly widens the gamut of tasks to 1.6k distinct NLP tasks.

2.6.2 RLHF and Learning from Preferences

Reinforcement learning from human feedback, or RLHF [34] is a method to optimize a language model against a set of expressed human preferences about its own output, to encourage the model to behave in a way that is consistent with human preference or expectation ("alignment"). RLHF involves several distinct phases: supervised fine-tuning (SFT), reward modelling, and finally proximal policy optimization or PPO. In the SFT stage, the pre-trained LM is first tuned directly on the desired input-output data, usually inputs and desired responses written directly by humans. The reward modelling stage involves humans judging the outputs of the model after the SFT stage, and expressing which of the two responses of a pair is preferred. From this, a reward model is trained, to be able to predict what the human preference would be given two responses. Finally, the LM is optimized against these preferences via PPO, an RL-based optimization method.

One thing to note from this entire process is the costly requirement of human interaction almost every step of the way. Humans are required to write the initial desirable responses for SFT, and are then needed again, for judging the model responses. The amount of preference data needed for the reward modelling step is not insignificant either. The entire process is quite costly, and as such, any methods to reduce the amount of human labor involved are highly desirable.

2.6.3 The Superficial Alignment Hypothesis

In light of what was just mentioned, in terms of reducing the amount of human labor required, one paper that aims to do this is known as LIMA [56]. LIMA aims to reduce the amount of human labor required for alignment by using only a small (1K) set of highly curated responses, and tuning exclusively via SFT. The authors of LIMA simultaneously propose a hypothesis known as the "Superficial Alignment Hypothesis". The Superficial Alignment Hypothesis states that the process of alignment, as performed by previous works, primarily concerns superficial stylistic elements of input and output, and does not teach the model any new knowledge or abilities that were not acquired already during pre-training. This hypothesis is introduced in support of the authors' endeavor to align a language model with only a small set of 1K input/output examples. The main relevant baseline the authors provide is measuring against Alpaca-65B, a model trained from LLaMA-65B by fine-tuning on 52K instruction-based input/output examples. The authors also investigate how the model behaves towards unsafe queries, showing that it does gain the ability to refuse to answer, since some refusals were included in the training dataset. This hypothesis is related to the work presented in Chapter 4, because if this hypothesis is true, then the implication is that we can potentially enforce this stylistic or surface-level alignment/instruction tuning via ICL as well, potentially with a very small set of task examples.

2.6.4 Prompting Models for Dialogue

One approach closely related to the content of Chapter 4 is to prompt a base language model to be a dialogue agent. Such approaches have been explored in the literature before [2, 16]. One major difference between the work presented in Chapter 4 and these existing works is that these works use prompting to induce dialogue-agent behavior, i.e. to produce helpful and harmless ("aligned") chatbots. Although zero-shot task generalization is potentially a component of the helpfulness axis of a generic dialogue agent, these works do not systematically investigate the ability for a prompted LM to generalize to novel unseen tasks like we do.

2.7 Evaluation of Generative Models

2.7.1 ROUGE metrics and their weaknesses

One of the main methods used to evaluate generative models, that arose from the development of translation and summarization systems, is a family of metrics known as ROUGE (Recall-Oriented Understudy for Gisting Evaluation). These metrics take a machine-generated output and a set of reference outputs written by humans, and evaluate based on different kinds of overlap between the machine output and the reference outputs. In this work, the main metric that is relevant is ROUGE-L. ROUGE-L evaluates a text based on the longest common subsequence (LCS) between it and the reference outputs. In this case, the LCS refers to the longest sequence of terms that appear in the same order, but are not necessarily contiguous, in both the reference texts and the machine output.

The main problem with ROUGE-L and other similar metrics is that they heavily rely on the quality and coverage of the set of reference outputs [1]. In other words, they rely on the idea that the humans writing the reference answers are able to adequately cover the space of all possible correct outputs to a given query. In most cases, given that cost is attempted to be minimized, this is far from true. There are many datasets where the set of reference answers is very small, if not only a single answer. As such, metrics like ROUGE-L may unfairly penalize generative models for answers that, were they to be evaluated directly by a human would be considered correct, but simply don't closely adhere to the wording provided by the reference answers they are being compared to.

2.7.2 Model-based Evaluation

An alternative strategy for evaluation of open-ended generation is model-based evaluation [8,12, inter alia]. In such a setup, the output of the model is passed to a (potentially much) larger LLM, potentially along with reference answers, and the larger model scores the generation. In one method, the larger model compares two generations head-to-head, and expresses a preference for one of the two, similar to the type of preference data that is generated by humans for RLHF. Another approach is to simply have the model score the answer quality directly. This method is also problematic, especially when used in conjunction with larger proprietary models whose pre-training data is

unknown. This results in there being many questions about the quality of the larger model's evaluations, as well as what biases may come into play during the evaluation. Nevertheless, model-based evaluation seems to be the predominant form of evaluation for open-ended generation for most new models being released.

Chapter 3

In-Context Learning for Text Classification with Many Labels

3.1 Introduction

3.1.1 Text classification with many labels

Text classification with large label sets is a relatively understudied NLP problem area. The majority of existing datasets involve very few classes, ordinarily under 10 and most often under 5. Looking at the GLUE benchmark, an extremely popular NLU benchmark that spans a variety of diverse tasks, the majority of the tasks are binary classification, with a few tasks being three-way classification or regression tasks instead.

Text classification into many classes in the past has mainly been motivated by practical considerations of building dialogue systems, where one component of a dialogue system is classifying the user's intent based on an utterance. One of the main ways to tackle such problems in the past was to use retrieval models or other traditional information retrieval (IR) approaches. With the advent of large language models (LLMs), which achieve high performance on a wide variety of tasks without specialized training, it is natural to wonder what kind of performance would be possible in such

classification tasks with an LLM. One of the most impressive abilities of LLMs is the ability to learn without parameter tuning, through in-context learning (ICL).

One of the main obstacles with applying LLMs to tasks involving classification with many labels is the limited context window these models have. Ordinarily with ICL, at minimum one example from each class is provided in-context to allow the model to make a choice between all the labels of the task. Because of this limitation, ICL has not been directly applied to these sorts of problems. In this work we relax this requirement, allowing the model to see only a subset of the most relevant labels for the given datapoint we are performing inference on.

3.1.2 Bypassing the context window constraint

In this Chapter, we demonstrate that not only can we apply LLMs to text classification problems with many labels, but in fact in doing so we can achieve state of the art (SoTA) performance. We test on intent classification (upwards of 50 classes) and fine-grained sentiment analysis (upwards of 25 classes). By coupling the LLM with an external pre-trained dense retriever model [19, 40], we can dynamically retrieve a set of examples to provide to the LM in-context, that reflects only the most relevant labels to the current example in the label space. Most existing work on augmenting LMs with retrieval models [36, 45] focuses on tuning the retrieval and/or LM. We demonstrate that even without tuning either, when the pre-trained models are strong enough we can still achieve SoTA across various tasks using ICL.

3.1.3 Novel contributions

The contributions of this work are:

1. We show that retrieval-augmented ICL is an effective way to tackle text classification tasks with many labels without additional tuning of either the retriever or the LM, either matching or outperforming fine-tuned adapter-based and contrastive-pre-training-based methods. Notably, truncating the dataset by showing only a subset to the LM at a time does not prevent us from

achieving SoTA performance, and allows us to apply LLMs to problems that they have not been applied to before,

2. We analyze ICL performance over different numbers of examples and demonstrate that larger models better are able to take advantage of more examples in-context than smaller models, which mostly plateau and/or see decreasing performance,

3. We perform several ablation studies to determine what aspects of the inputs and outputs the model is using for ICL. Certain recent works investigating ICL [31, 37] have recently called into question how much models are actually "learning" with ICL and what they are learning from. We ablate three different elements (semantic label names, correct input-output correspondences, and semantically similar demonstrations to the current input). Contrary to this emerging literature, our experiments demonstrate that they are all used to varying degrees, depending on the dataset and domain.

4. We experiment with a large variety of example selection methods, and show that in fact ordinary nearest-neighbor retrieval is the most effective out of all the different selection methods tested, across multiple datasets,

5. We show that least-to-most ordering is the most effective way to order retrieved examples in-context, across multiple datasets and domains,

6. We show that neural retrievers provide a significant boost over BM25, a traditional term-based retriever,

7. We show that fine-tuning the retriever model does not provide consistent improved performance, contrary to intuition,

8. We show that more powerful neural retrievers also do not consistently improve performance,

9. We show that RLHF-aligned models perform worse at few-shot retrieval-augmented classification than their equivalent base LMs

3.2 Datasets

In this section, a brief overview of the datasets used in this Chapter is given. Four datasets were used in total in this work: HWU64 [25], BANKING77 [6], CLINC150 [21], and GoEmotions [10]. Out of these, 3 are in the broader area of intent detection, and 1 is in the category of emotion classification. In this section each dataset is briefly described, and some sample datapoints are provided.

3.2.1 HWU64

HWU64 [25] is an intent detection dataset composed of 64 distinct intent classes, created to mimic real-world interactions between users and voice assistant systems (e.g. Alexa, Siri). The intent classes in HWU64 are diverse in nature, and can be broken down into groups like:

1. Querying for information: this category includes intents that are knowledge-seeking, such as asking for the weather, the news, or stock prices.

2. Personal assistant tasks: this category includes intents related to tasks that a personal assistant would engage in, such as creating calendar events or inviting people to meetings.

3. Internet of things (IoT tasks): this category includes intents related to controlling IoT devices, such as coffee makers, robotic vacuum cleaners, or controllable lights.

4. Music-related tasks: this category includes intents relating to playing music, pausing, searching for specific tracks, among others.

5. Travel: this category includes intents relating to travel, such as booking train or flight tickets, booking accommodations, or checking travel conditions.

The dataset includes several other categories similar to the above. Some datapoint samples are provided in Table 3.1.

Table 3.1: Sample datapoints from HWU64

Text	Gold label
get me a seat on the next train going to new york	transport_ticket
mute yourself until five pm	audio_volume_mute
is i. b. m. up today	qa_stock
should i wear a hat today	weather_query
search cost for amtrak luxury to los angeles	transport_query
i've cancelled the order placed at mcd did it go through	takeaway_query
what's playing at brea plaza five	recommendation_movies

Table 3.2: Sample datapoints from BANKING77

Text	Gold label
I didn't get all the cash I requested for at the ATM	wrong_amount_of_cash_received
From where can I withdraw?	atm_support
I need help changing my last name on my account	edit_personal_details
How do people send me money?	receiving_money
Is a non-electronic card available as well	order_physical_card
When I travel, what will it cost to switch for my currency?	exchange_charge
i live in the US can i still get a card?	country_support

3.2.2 BANKING77

The BANKING77 dataset [6] is a more fine-grained and domain-specific intent detection dataset than HWU64. The datapoints in BANKING77 are specifically related to the domain of banking. The creators of the dataset categorize utterances into 77 banking-related intents, covering a wide range of banking-related tasks, such as account management, card payments, exchange rate queries, among others.

Some datapoint samples are provided in Table 3.2.

3.2.3 CLINC150

CLINC150 [21] is the third and final intent classification dataset tested. CLINC150 spans 150 different intent classes, grouped into 10 different domain categories. Domains include travel, work, banking, and small talk.

Some datapoint samples are provided in Table 3.3.

Table 3.3: Sample datapoints from CLINC150

Text	Gold label
i want to eat mediterranean fare with at least four stars, near me	`restaurant_suggestion`
tell my bank i'll be in cuba beginning the 2nd	`travel_notification`
i don't have butter can i use oil	`ingredient_substitution`
will i be charged if i use my card in japan	`international_fees`
how do i say thank you in japanese	`translate`
how much of my time off have i used	`pto_used`
what do i do to fix a dead car battery	`jump_start`

3.2.4 GoEmotions

GoEmotions [10] is a dataset for fine-grained emotion classification. This dataset goes beyond traditional emotion classification datasets, which typically are either 2-way (positive, negative) or 3-way (positive, negative, neutral) classification. The authors at Google produced a dataset with 28 different fine-grained emotion categories. These include similar emotions such as anger, annoyance, disapproval, disgust, embarassment, to name a few. Given the fine-grained nature of the categories, and their frequent similarity or possibly even subjective nature, GoEmotions is a very challenging dataset, and as such is a great testbed for retrieval-augmented ICL. The dataset is extracted from comments from the Reddit social media platform, giving it diverse and realistic utterances. These utterances were then annotated by human raters.

Some sample datapoints are provided in Table 3.4.

3.3 Nearest Neighbor In-Context Demonstration Selection

Retrieval-Augmented ICL: Our setup assumes N classes (unique labels) with K examples in each class. Each example is composed of an `(input, label)` tuple. We assume that we have a limited number of examples M to fit in the prompt, based on the model's context length. M can be fixed or based on "saturating" the prompt greedily by selecting examples until we run out of room in the context window. From our total pool of examples of size $N \times K$, we retrieve the M examples using the cosine similarity values given by our retrieval model. Having retrieved our M

Table 3.4: Sample datapoints from GoEmotions

Text	Prediction LLaMA-2-70B	Gold label
Lmao the brigading is real	amusement	amusement
Enjoy the void	neutral	neutral
I really relate to this.	realization	approval
This is the only viable way out of Brexit.	optimism	approval
want* a source on that, sorry.	desire	remorse
I didn't know that, thank you for teaching me something today!	gratitude	gratitude
Well it obviously helps you rationalize your total unwillingness to take action to make the world a better place. I hope that you grow past that.	sadness	admiration
Damn, we need healthy PGs.	sadness	annoyance
Welcome to The Church of Jesus Christ of Latter Day Saints, where families can be SEPARATED forever	sadness	gratitude

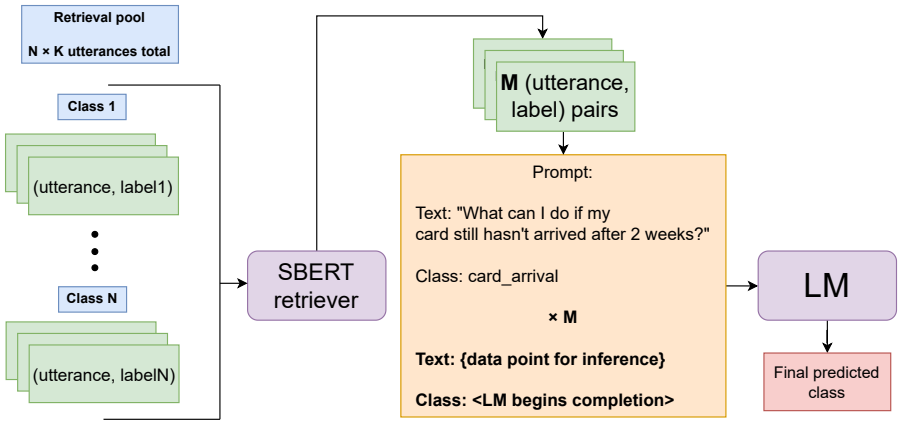

Figure 3.1: Complete pipeline for intent detection with retrieval-augmented in-context learning

examples, we then produce the prompt by concatenating the `(input, label)` tuples in a set prompt format (see Figure 3.1), similar to existing in-context learning setups. The final prediction is then taken from the LM by having it produce a continuation based on our prompt. A full visual description of the retrieval process is visible in Figure 3.1.

Retrieval model: The retrieval model used is a Sentence-BERT model trained in a Siamese dual-network setup to be able to retrieve text based on cosine similarity of the embedding vectors it produces, described in [41]. The model we use is a contrastively trained model which has been

pre-trained on a massive generic dataset of text pairs. We use the retrieval model as-is in all experiments. Cosine similarity is used to retrieve examples from the retrieval pool of examples (tested in 5-shot and 10-shot scenarios, signifying the number of examples from each class in the retrieval pool).

3.4 Experimental Setup

Specific retrieval model: For our sentence encoder/retriever, we use the SentenceTransformers library [40], and use the pre-trained "all-mpnet-base-v2" model (a 110M parameter model pre-trained on over 1 billion training pairs). The SetFit results are based on contrastively tuning the same pre-trained model trained by Microsoft through the Setfit library[1].

Prompt saturation: The number of examples that fit in-context when greedily filling the context window depends on the specific dataset. For the intent detection datasets, this number was around 110 examples. For GoEmotions, this number was around 70 (140 using the full 4K context length of the LLaMA-2 models).

Splits: For the intent detection experiments, to allow for direct comparison with previous works, we use the same 5-shot and 10-shot sets as DialoGLUE [28]. Experiments are run 3 times and the accuracies are averaged, except the zero-training LLM setups, which are deterministic. For the GoEmotions experiments we average the results across 3 different random 10 and 5-shot splits, as no pre-existing few-shot splits exist. The GoEmotions experiments are composed of the subset of GoEmotions data (84% of training set, 85% of testing set) where the there is only one emotion label, to avoid issues of enforcing an ordering on a linearized version of multiple labels in sequence, as well as to mimic the single-label intent detection datasets setup more closely. Default library parameters were used.

[1] https://github.com/huggingface/setfit

Computing Hardware and model differences: All experiments were performed on a single A100 80GB GPU, except those with OPT 175B, which were performed with 8 A100 GPUs. For LLaMA 65B and 70B 8-bit quantization was used. The main difference between the OPT and LLaMA models is the amount of pre-training data used. The LLaMA models were trained on 1T-1.4T tokens, while the OPT models were only trained on 180B tokens (see [54] and [47] for more details). LLaMA-2 models were trained on 2T tokens.

3.5 Nearest Neighbor Results

An overview of the results using nearest neighbor retrieval and ICL are provided in Tables 3.5 and 3.6.

Table 3.5: Intent classification accuracy for retrieval+ICL and baseline methods. All retrieval+ICL results are with 20 in-prompt examples unless otherwise specified. The retrieval/training dataset size is given by the second row of the header (10-shot is 10 examples per class, 5-shot is 5).

Model	**BANKING 77**		**HWU 64**		**CLINC 150**	
	5-shot	**10-shot**	**5-shot**	**10-shot**	**5-shot**	**10-shot**
SBERT 1-NN	78.41	85.39	69.89	75.46	82.51	84.84
ConvFit (reported)	-	87.38	-	85.32	-	92.89
SetFit	79.89 $\pm$ 0.14	84.51 $\pm$ 0.60	78.38 $\pm$ 0.73	83.35 $\pm$ 0.57	88.68 $\pm$ 0.20	90.67 $\pm$ 0.29
DeBERTa (Pfeiffer)	81.47 $\pm$ 1.6	88.41 $\pm$ 0.19	79.80 $\pm$ 0.81	86.93 $\pm$ 0.052	91.86 $\pm$ 0.66	95.05 $\pm$ 0.33
OPT 13B	81.23	85.65	78.90	83.64	85.27	89.24
OPT 175B	81.30	86.14	83.74	84.94	90.96	93.09
LLaMA 7B	84.42	87.63	85.87	87.55	88.58	91.73
LLaMA 65B	87.73	90.71	89.03	90.06	91.89	94.47
LLaMA 2 7B	86.40	89.45	87.55	87.82	94.13	95.20
LLaMA 2 7B 4K	85.91	89.48	87.17	90.33	95.35	96.02
LLaMA 2 70B	87.56	90.58	88.20	89.77	96.42	97.13
LLaMA 2 70B 4K	**88.96**	**92.11**	**90.61**	**91.73**	**97.56**	**98.18**

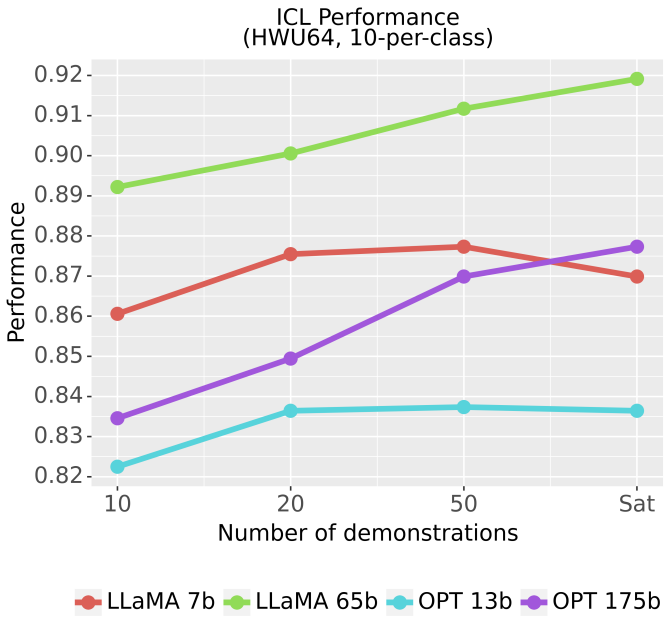

Figure 3.2: HWU performance as a function of the number of examples in prompt. The x-axis scale is non-linear, meaning that there are diminishing returns with more examples. "Sat" (saturated) indicates filling the prompt greedily until the max length is reached.

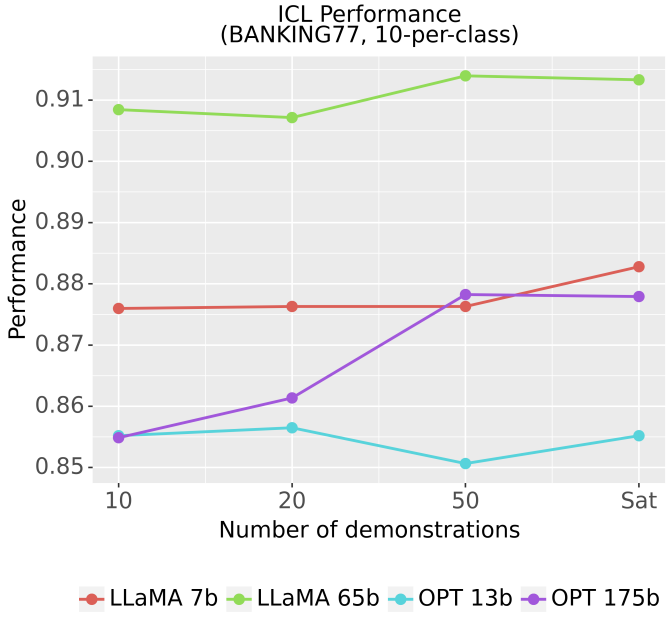

Figure 3.3: BANKING performance as a function of the number of examples in prompt. The x-axis scale is non-linear, meaning that there are diminishing returns with more examples. "Sat" (saturated) indicates filling the prompt greedily until the max length is reached.

Table 3.6: Sentiment classification macro F1 score (following prior work) over 3 random splits for retrieval+ICL and baseline methods. All retrieval+ICL results are from saturating the prompt with in-prompt examples (with a 2K prompt length unless otherwise specified). The retrieval/training dataset size is given by the second row of the header (10-shot is 10 examples per class, 5-shot is 5). +Neut refers to the case where the "neutral" class (lack of emotion) is included in the dataset.

Model	**GoEmotions**			
	5-shot	**10-shot**	**5-shot +Neut**	**10-shot +Neut**
SBERT 1-NN	9.48 ± 0.58	11.02 ± 1.0	7.55 ± 0.79	8.38 ± 0.48
SetFit	25.44 ± 4.5	34.69 ± 3.6	21.40 ± 3.18	27.78 ± 0.73
DeBERTa (Pfeiffer)	18.43 ± 2.9	32.33 ± 0.77	13.86 ± 1.49	25.42 ± 1.9
LLaMA 7B	-	-	22.99 ± 0.64	24.61 ± 0.47
LLaMA 65B	-	-	24.31 ± 0.73	25.63 ± 0.86
LLaMA 2 7B	29.60 ± 1.5	-	23.78 ± 1.1	24.75 ± 0.43
LLaMA 2 7B 4K	28.01 ± 1.2	30.33 ± 1.64	23.79 ± 1.9	23.57 ± 0.52
LLaMA 2 70B	**36.14** ± 1.7	**37.81** ± 1.3	24.20 ± 0.13	25.29 ± 0.42
LLaMA 2 70B 4K	-	37.17 ± 0.37	**28.26** ± 0.19	**29.10** ± 0.68
LLaMA 2 70B 4K Retrieval w/o Neutral	-	-	-	28.95 ± 0.52

3.5.1 Discussion: Small models cannot use long contexts as effectively as large models

One trend noticeable from the performance graph as a function of the number of examples for HWU (see Figure 3.2) is that small models seem to be unable to use more examples as effectively as large models. The smaller OPT model is unable to effectively make use of the entire context window when it is filled and remains at relatively low performance. In contrast, OPT 175B shows continual improvement when more examples are added. A similar trend is visible for the LLaMA models, where the performance of the 7B model does not change significantly (see Figure 3.2), but the 65B model is able to continuously improve. The smaller models either level off (OPT-13B) or lose performance (LLaMA-7B). In the 4K full context window settings for LLaMA-2, the difference between model scales is even more apparent (Tables 3.5 and 3.6). We see the small model showing inconsistent use of the longer contexts; sometimes improving, but mostly staying the same or

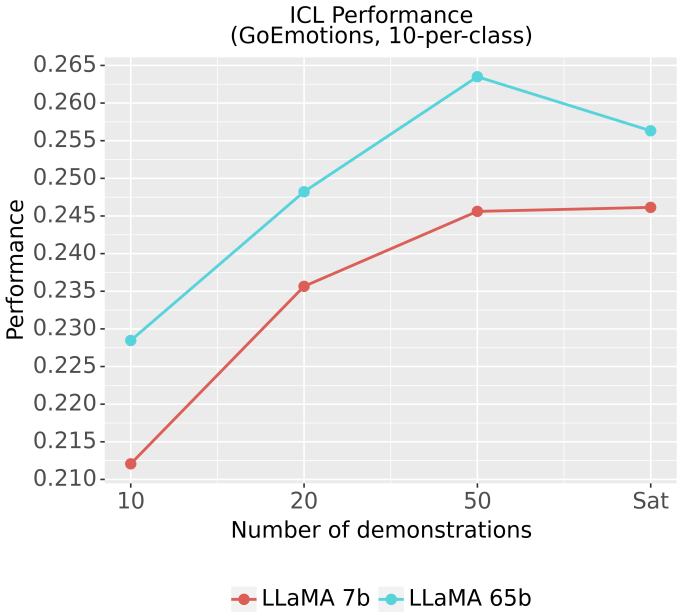

Figure 3.4: GoEmotions performance as a function of the number of examples in prompt. The x-axis scale is non-linear, meaning that there are diminishing returns with more examples. "Sat" (saturated) indicates filling the prompt greedily until the max length is reached.

worsening performance. Meanwhile, the large model consistently improves with the full context in almost all cases.

3.6 Alternative Selection Approaches

In addition to regular nearest-neighbor retrieval, we additionally experiment with several other retrieval strategies.

3.6.1 Upper-bounded Number of Classes

We try doing the pure nearest example approach, but with a restriction to a fixed M number of classes represented in the prompt (i.e. as we are adding examples, if we reach a certain M number of classes represented in the prompt, we stop adding examples of other classes, and just fill the prompt with examples of the first M classes, in order of similarity). This was to see if the LM potentially was having issues handling examples of too many classes in the prompt. The number of

classes (not demonstrations) returned by the retriever as a function of the number of demonstrations retrieved is shown in Figure 3.8. Once the number of demonstrations reaches 50, the number of classes retrieved is 15 or more. The reasoning behind trying this selection approach was that this constitutes too many classes for the LLM to be able to reliably distinguish in a single prompt. The algorithm is described in Algorithm 1.

Algorithm 1 Max Class Selection Algorithm

Require: Set of demonstrations S
Require: Query datapoint Q
Require: Target number of demonstrations N
Require: Max number of classes represented in the prompt M

procedure MAXCLASSSELECTION(Q, S, N, M)
    # demonstration set for prompt that we will build
    $D \leftarrow \emptyset$
    # set of classes we have already seen
    $C \leftarrow \emptyset$
    # first, order demonstration set by similarity to query
    $S \leftarrow$ORDERSIM(S, Q)
    $i \leftarrow 0$
    while LEN(D) $< N$ **do**
        if LEN(C) $< M$ **then**
            $D \leftarrow S[i]$
            $C \leftarrow$ CLASS($S[i]$)
        else
            # If we have reached the max number of classes,
            # only allow the example into D if its class is already in C
            if CLASS($S[i]$) $\in C$ **then**
                $D \leftarrow S[i]$
            end if
        end if
        $i \leftarrow i + 1$
    end while
    Return D
end procedure

3.6.2 Deduplication

We try a "deduplicative" approach to try a more diverse prompt, where an example would not be added to the prompt demonstration pool if it was too similar to an existing example in the pool. This

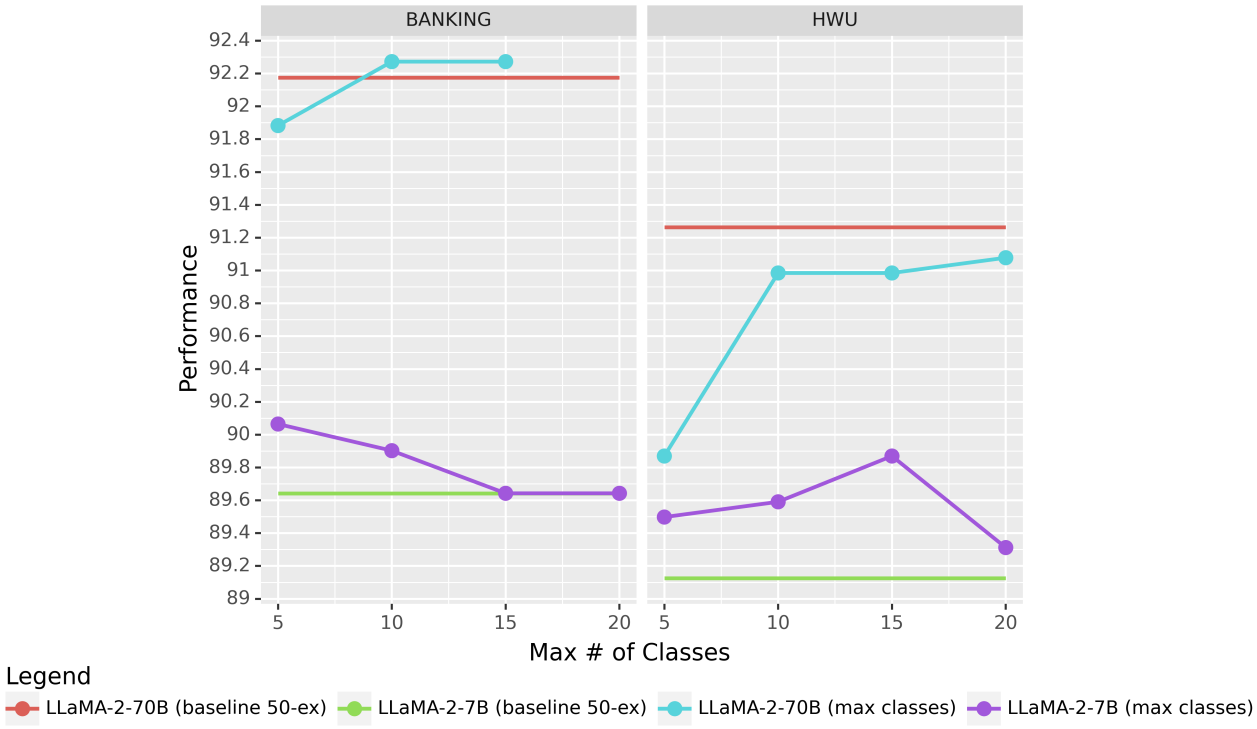

Figure 3.5: The max classes selection approach, showing performance as a function of the number of classes ceiling.

algorithm is shown in detail in Algorithm 2. Worth noting is the fact that this algorithm can possibly fail, if we set too low a threshold. The thresholds tested empirically all succeeded in building a demonstration set with the target number of demonstrations.

3.6.3 Class Balancing

We try "balancing" the classes in the prompt, i.e. giving a fixed N examples from each of the nearest M classes, where "nearest M classes" is defined by each class's nearest example to the input instance. This algorithm is described in detail in Algorithm 3. The use of a stack makes it so that N does not need to be evenly divisible by M. There are two supplemental algorithms used in the main algorithm (Algorithms 4 and 5).

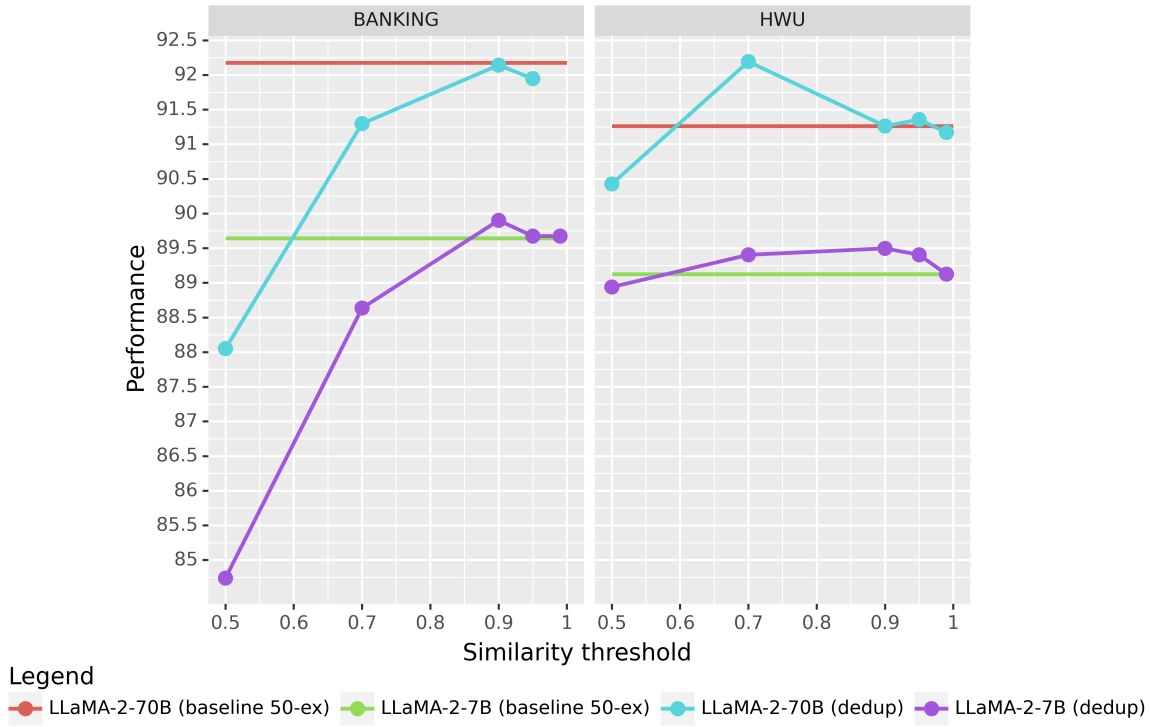

Figure 3.6: The deduplication selection approach, showing performance as a function of deduplicative threshold.

3.6.4 Discussion: Alternative demonstration selection approaches mostly fail

Most of the alternative demonstration selection approaches fail. Specifically, in the few cases where they show stronger performance than the default nearest ordering, the performance boost is relatively slight (in the best case, around 1%). The cases where the alternative selection methods improved performance seem to universally apply to the smaller 7B model and not to the 70B model. Furthermore, all of the alternative demonstration methods require tuning an additional hyperparameter, making them less powerful than simple nearest example selection.

Maximum number of classes selection approach: This approach yielded some small improvements. (see Figure 3.5). For the large model (70B), it either does not produce better results (BANKING), or produces worse results (HWU) than the baseline (pure nearest neighbor). However, in the 7B case, there seems to be some limited improvement (around 0.5%-0.6%). The improvement

Algorithm 2 Deduplicative Selection Algorithm

Require: Set of demonstrations S
Require: Query datapoint Q
Require: Target number of demonstrations N
Require: Deduplicative threshold t

procedure DEDUPLICATIVESELECTION(Q, S, N, t)
    # demonstration set for prompt that we will build
    $D \leftarrow \emptyset$
    # first, order demonstration set by similarity to query
    $S \leftarrow$ORDERSIM(S, Q)
    $i \leftarrow 0$
    while LEN(D) $< N$ **do**
        for $d \in D$ **do**
            # If the candidate demonstration is too similar to any previous demonstrations
            # we have selected, do not add it to D
            if SIM(d,$S[i]$) $> t$ **then**
                skip $\leftarrow$ true
            end if
        end for
        if $\neg$ skip **then**
            $D \leftarrow S[i]$
        end if
        $i \leftarrow i + 1$
    end while
    Return D
end procedure

however is marginal, which seems to indicate that the retriever is returning a number of classes that both the 7B and 70B can relatively reliably handle. In other words, the 15+ classes retrieved at 50 demonstrations shown in Figure 3.8, in fact not so many that the models completely get overwhelmed. The 7B model benefitted slightly from the restriction to a maximum number of classes, but generally both models performed OK, with the larger model performing significantly worse with the cap at 5 classes in the HWU case. This is likely additional evidence that the large model specifically is able to make full use of the variety of demonstrations (many classes), so much so that capping the number of classes so low significantly impedes its performance, while the small model shows improved performance at this cap in contrast.

Algorithm 3 Class Balancing Selection Algorithm

Require: Set of demonstrations S
Require: Query datapoint Q
Require: Target number of demonstrations N
Require: Number of Classes to Represent Equally in Prompt M

procedure CLASSBALANCINGSELECTION(Q, S, N, M)
    # demonstration set for prompt that we will build
    $D \leftarrow \emptyset$
    # set of classes we have already seen
    $C \leftarrow \emptyset$
    # first, order demonstration set by similarity to query
    $S \leftarrow$ORDERSIM(S, Q)
    $C_{ord} \leftarrow$ ORDERCLASSESBYNEAREST(S)
    # Sort S into a hash table with the key as the class and the value as the
    # list of demonstrations of that class sorted by similarity to the query
    $S_{classes} \leftarrow$ CLASSESDICTORDERED(S)
    $i \leftarrow 0$
    # the current class we are taking from
    $c \leftarrow C_{ord}[0]$
    while LEN(D) $< N$ **do**
        # treat $S_{classes}$ as a stack, pop the next most similar
        $D \leftarrow$ POP($S_{classes}[c]$)
        $i \leftarrow i + 1$
        $c \leftarrow C_{ord}[i \mod M]$
    end while
    Return D
end procedure

Algorithm 4 Class Balancing Selection Supplemental Algorithm 1

Require: Set of demonstrations already ordered by similarity S

procedure ORDERCLASSESBYNEAREST(S)
    $C_{ord} \leftarrow \emptyset$
    for $d \in D$ **do**
        if CLASS(d) $\notin C$ **then**
            $C_{ord} \leftarrow$ CLASS(d)
        end if
    end for
    Return C_{ord}
end procedure

Algorithm 5 Class Balancing Selection Supplemental Algorithm 2

Require: Set of demonstrations already ordered by similarity S

procedure CLASSESDICTORDERED(S)
    $S_{classes} \leftarrow \emptyset$
    # Iterate over all classes represented in S
    for do $c \in$ CLASSES(S)
        $S_{classes}[c] \leftarrow \emptyset$
        for do $d \in S$
            if CLASS(d) $= c$ **then**
                $S_{classes}[c] \leftarrow d$
            end if
        end for
    end for
    Return $S_{classes}$
end procedure

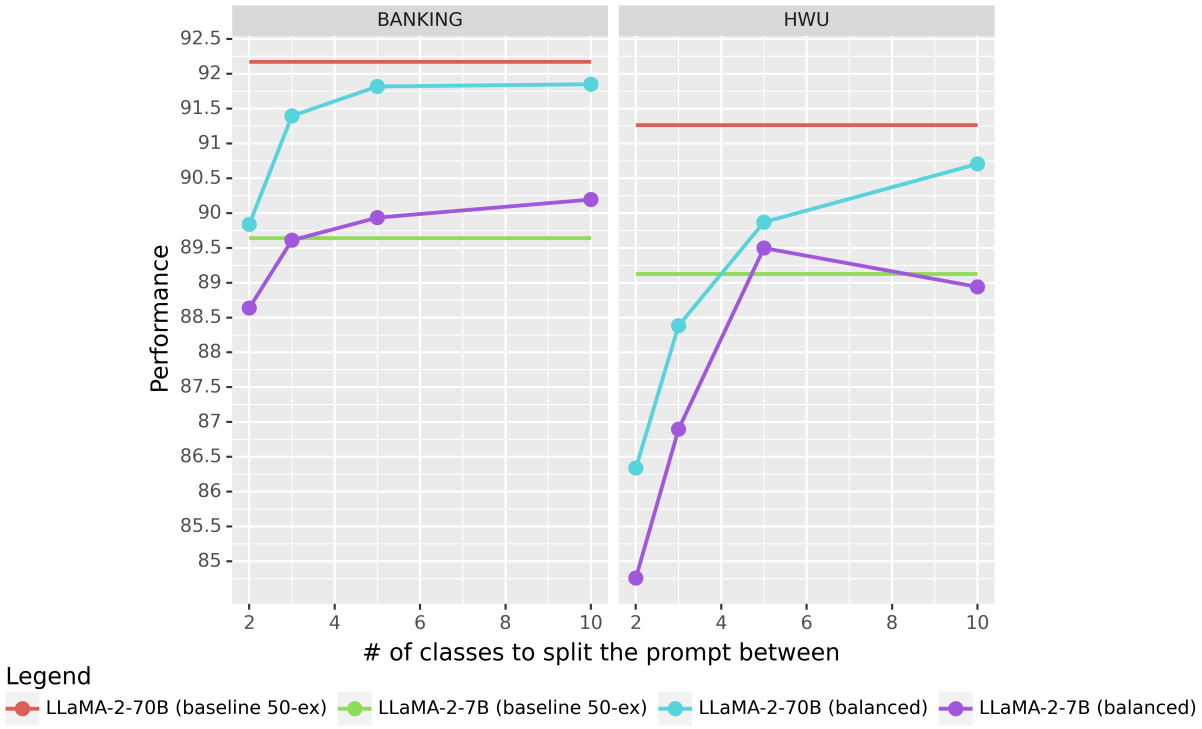

Figure 3.7: The class balancing selection approach, showing performance as a function of the number of classes the prompt is split between.

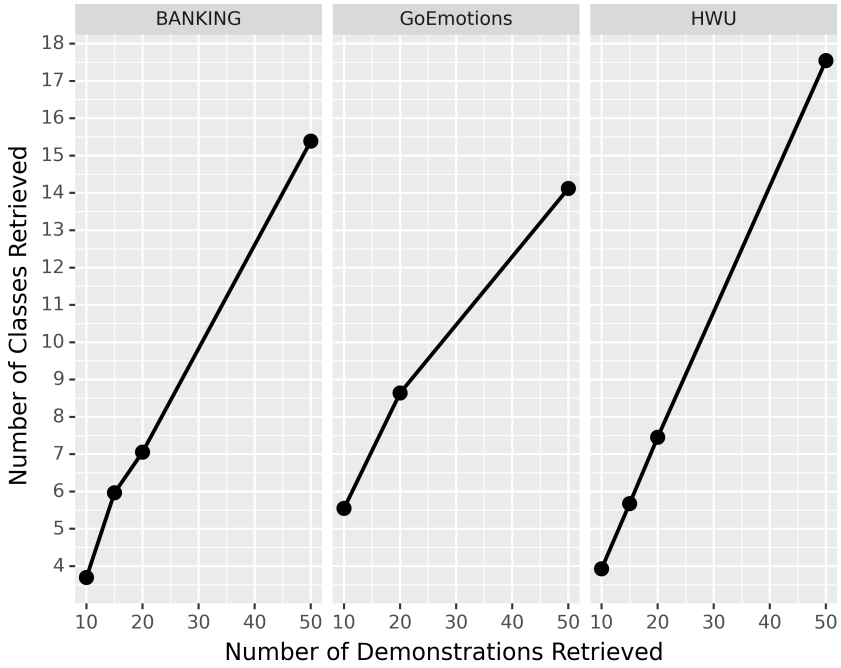

Figure 3.8: The number of class represented in the retrieval pool as a function of the number of demonstrations retrieved.

Deduplication approach: This approach yielded the most improvement out of all the selection approaches tested, although the improvements are still marginal (see Figure 3.6). At low similarity thresholds with BANKING (i.e. the most stringent deduplication), both 7B and 70B perform significantly worse than the baseline. With HWU at a 0.9 similarity threshold, with the 7B model, there seems to be a slight improvement of around 0.3%. The highest improvement came from the 70B model on HWU at a 0.7 similarity threshold, where performance went from 91.25% to 92.25%, a difference of 1%. However, this effect was present only at this specific threshold value, and therefore it is difficult to extract any generalizable conclusions from this.

Classes equally represented in prompt (balancing) approach: This approach performed worse in all cases for the 70B model (see Figure 3.7). However, for the 7B model we see around 0.5% improvement in BANKING when splitting the prompt among 10 classes, and around 0.25% improvement for HWU when splitting among 5. It seems that removing the information about the

distribution of classes in the embedding space neighborhood of the query (which is effectively what this approach does, by splitting the prompt among the classes equally) harms performance.

3.7 Ablation Experiments

Several ablations studies are done to test what aspects of the retrieved examples the LLM is using to make the predictions. The ablation studies were done on a random split of the HWU dataset and the GoEmotions dataset. Ablation results for HWU are shown visually in Figure 3.9 and for GoEmotions in Figure 3.10.

1. **Obfuscated labels:** We change all the class names to randomly set enumerated names ("Class 1", "Class 2", etc.). The intent is to disentangle the model's use of prior (pre-training) knowledge to perform the task (based on the semantic content of the label names) from the input-output provided in the prompt.

2. **Resampled in-context examples:** To test if similarity between the demonstrations provided in the prompt and the current input example is actually necessary for effective performance. By resampling from the classes initially retrieved by the retriever model, we preserve the distribution of labels but change the input demonstrations themselves so that they are no longer the nearest in the embedding space for each class.

3. **Shuffled labels:** Similarly to [31], after the retrieval step we shuffle the correspondence between the inputs and labels of the retrieved examples, such that inputs are matched randomly from the set of labels the inputs originally belonged to. The intent of this ablation is to examine if the model requires correct input-label correspondences (something that [31] calls into question), or if the model is simply using structural (e.g. prompt format) and distributional (e.g. the distribution of labels in the prompt) elements to produce a prediction.

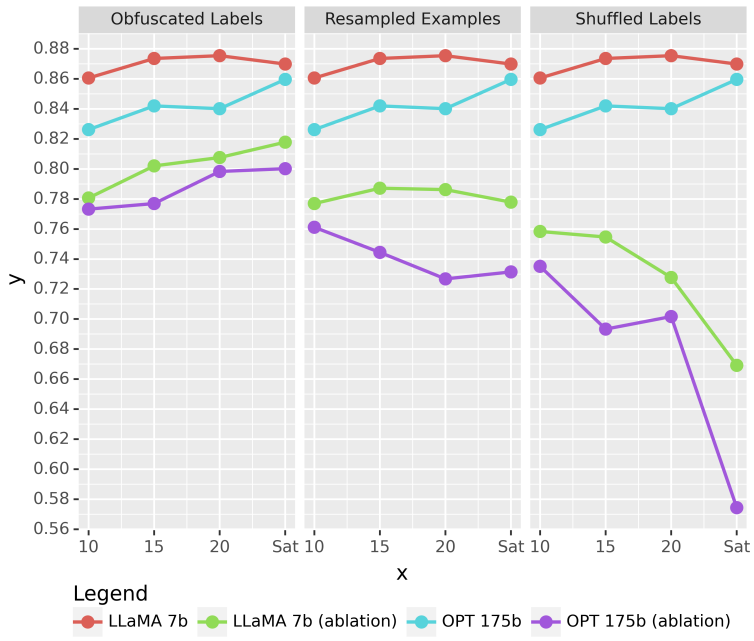

Figure 3.9: Classification accuracy for three ablations for HWU64: obfuscated labels (left), resampled in-context examples (center), shuffled labels (right).

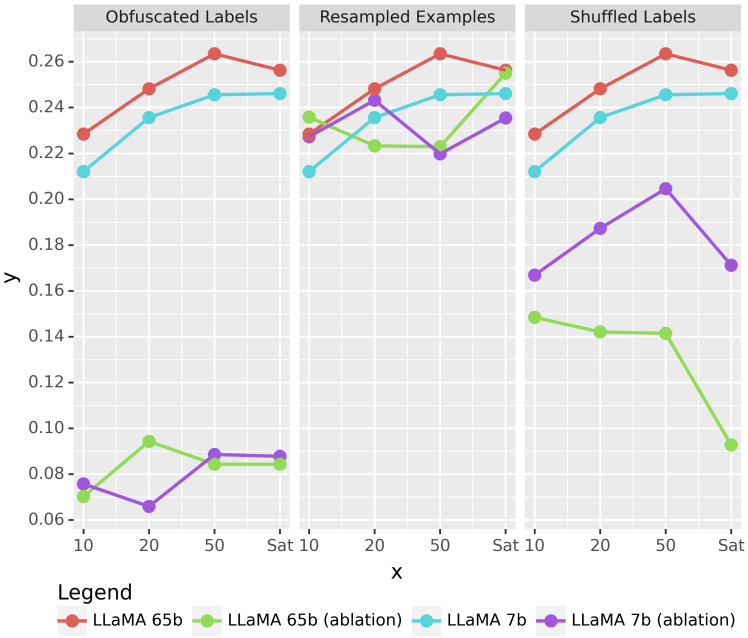

Figure 3.10: Classification accuracy for three ablations for GoEmotions: obfuscated labels (left), resampled in-context examples (center), shuffled labels (right).

3.7.1 Discussion: Similarity to current datapoint matters for intent classification

In the resampling ablation for HWU (see Figure 3.9) we see that resampling from the initial class distribution provided by the retriever model damages the performance across both OPT 175B and LLaMA 7B. This supports the strong performance numbers of the LLMs, showing that the similarity between in-context demonstrations and the current input matters. This implies that the LM is doing more than just selecting the most common class or just using the shortlist of class labels from the full set of classes to select in a more zero-shot fashion. One interesting difference to note is that OPT 175B, the larger model, shows a larger drop from the resampling as the number of in-context demonstrations increases, compared to LLaMA-7B, whose performance stays roughly constant (but lower than non-resampled). This may indicate that the LLaMA models with their additional training data are more robust to the resampling process, due to stronger pre-training knowledge and/or more robust performance overall. In the case of GoEmotions, we see almost no variation with resampling, showing that similarity to the input example is less influential, though the ordering of the examples relative to each other does seem to make a difference for the 7B model (Table 3.7).

3.7.2 Discussion: Semantically significant label names matter greatly for sentiment classification

In the obfuscation ablation (see Figure 3.9), we see that all models are hurt by obfuscating label names. We see however that models are still able to learn to perform the task effectively, and in fact show similar improvement curves with increasing number of examples, just with a lower starting performance. This demonstrates that the semantic content of the labels is significantly useful to the models but simultaneously it is not integral to performing the task, which can also be done without semantically significant labels. In the case of GoEmotions, we see that the obfuscated labels particularly hurt the model, bringing it down significantly. It seems to be the case that the class names are integral to performance, but at the same time more examples are still helpful to the model, as in the 4K context window it still sees improved performance.

3.7.3 Discussion: Input-label correspondence matters for all datasets

Shuffling the input-label correspondence is the ablation in which we see the performance of all the models decrease the most in the intent detection case (see Figure 3.9). Specifically, we see that the performance drop is proportional to the number of examples (more shuffled examples brings a larger drop). That being said, it is noteworthy that the performance of both models in this shuffled regime is still significantly above random chance for every number of demonstrations shown, implying perhaps that the LM's prior knowledge based on the label names is still contributing significantly to performance. As such, this can be seen as a confirmation of the results of [31], where the authors show that randomizing labels does not drastically hurt performance. In our case the impact on performance is apparent, but since the model does not go down to random chance, it is still leveraging its knowledge and ignoring the incorrect input-output correspondence given in-context.

In all 4 datasets (intent classification and GoEmotions), shuffling the labels hurts the large model more in particular. This aligns with the results of [52], whose authors show that larger models are more able to learn perturbed input correspondences than smaller models, which manifests in this experiment as lower performance. In other words, the larger model is trying to learn the perturbed input correspondence, and thus losing more and more performance with more examples, while the smaller model is able to more effectively ignore the perturbation.

3.8 Retriever and LM Generalization

One interesting result from our experiments is the fact that generic retrievers seem to be able to quite effectively generalize across domains and tasks. Using the same exact retriever model across 3 different intent detection datasets (which according to the taxonomy of [18] constitutes cross-task generalization) as well as a sentiment classification dataset (according to the previous taxonomy, a cross-domain generalization) demonstrates SoTA or better performance in almost all cases. The distribution shift locus for both the retriever and the language model generating the final prediction, is from pretraining to testing time, as both the retriever and language model are pre-trained on massive generic data before being tested in a zero-shot setting.

3.9 Best ordering for demonstrations

Table 3.7: Comparison of select LLaMA and OPT model sizes vs. prompt orderings on intent detection datasets (20 examples in prompt, 10-shot), random split. MTL is most-to-least similar and LTM is the inverse.

Model	BANKING		HWU		CLINC		GoEmotions		
	MTL	LTM	MTL	LTM	MTL	LTM	MTL	Random	LTM
OPT 13B	73.64	**85.65**	76.39	**83.64**	81.11	**89.24**	-	-	-
ShearedLLaMA 2.7B	77.69	**86.23**	80.29	**84.20**	87.82	**92.44**	-	-	-
LLaMA 7B	83.64	**87.63**	86.99	**87.55**	90.20	**91.73**	15.91	20.89 $\pm$ 0.85	**23.58**
LLaMA 65B	88.08	**90.71**	89.03	**90.06**	93.47	**94.47**	-	-	-

One thing we see from the ordering experiments (Table 3.7) that goes against previous literature on the subject [23] is that the least-to-most (LTM) similar ordering seems to outperform other orderings across all datasets and domains tested. Previous literature found that the best ordering may be dependent on the specific dataset tested, but in our experimentation LTM ordering wins every time. Additionally, order sensitivity does *not* seem to be correlated with model size exactly (see Table 3.7). ShearedLLaMA is significantly smaller than OPT 13B but exhibits significantly less variation. The correlation rather seems to be with model performance (i.e. the stronger models are less sensitive to ordering).

3.10 Using more powerful neural retrievers

Results using a more powerful neural retriever (GTR-XL) are shown in Table 3.8. GTR-XL is a T5-based retriever with 1.24B parameters.

Looking at the T5 (GTR-XL) experiment results (Table 3.8), we see that more powerful neural retrievers do not always improve performance. This is interesting as it implies the bottleneck is not the retriever, but rather the LM making the final prediction, and that there is not a consistent rule of more similar examples in the context window leading to better performance. In other words, this seems to indicate that the performance of all of the neural retrievers is "saturated" with regards

Table 3.8: Comparison vs. GTR-XL Retriever (50-ex)

Model	BANKING	HWU
	10-shot	10-shot
LLaMA-2-7B (mpnet)	89.64	89.13
LLaMA-2-7B (GTR-XL)	89.19	88.66

to the tasks we evaluated, in that they are all equally effective at retrieving demonstrations for the language model.

3.11 Using classical retrievers

In this section we compare the SentenceTransformers neural BERT-based retriever against a classic Okapi-BM25 retriever on the HWU64 dataset. In Table 3.9 we can see that the traditional retriever performs measurably worse than the neural SentenceTransformer retriever, indicating that semantically-aware neural retrieval is necessary for high performance.

Table 3.9: Comparison vs. Classical (BM25) Retriever

Model	BANKING	HWU
	10-shot	10-shot
LLaMA-2-7B (neural)	89.45	87.82
LLaMA-2-7B (BM25-Okapi)	84.90	84.02

The BM25 retriever tested is significantly worse than the neural retrievers tested. The difference between classical and neural retrieval is discussed extensively in Chapter 2. This was an expected result, as the BM25 retriever is exclusively surface-form based, and therefore cannot handle the use of synonyms in the utterance vs. the query. Additionally, it is a known weakness of BM25 and other similar term-based retrieval methods that include a term frequency component (TF) that they are not as effective for short texts. The reason for this is that in short texts, especially as short as in this work, terms are most likely to appear only once or twice, effectively reducing the ability of BM25 to differentiate documents based on this term. In addition, the inverse document frequency

(IDF) component becomes less informative, as the probability of having unique terms decreases. Along the same lines, all the tasks tested involve dataset with relatively simple, everyday language. This kind of language involves a high frequency of common terms, again reducing the effectiveness of both the TF and IDF components. In general, BM25 was designed for long documents. The type of data we are dealing with requires the ability to grasp the semantics of the text to some extent, which is only possible with a neural retriever.

3.12 Fine-tuning retrievers

Table 3.10: Comparison of Models with Fine-tuned Retriever (20 examples in prompt), compared against non-fine-tuned performance

Model	**BANKING**	**HWU**	**CLINC**
	10-shot	**10-shot**	**10-shot**
SBERT KNN	87.40 ± 0.21	83.05 ± 0.47	91.48 ± 0.13
vs. frozen	+ 2.0%	+ 7.6%	+ 6.64%
OPT 13B	87.71 ± 0.18	83.83 ± 0.83	91.83 ± 0.22
vs. frozen	+ 2.06%	+ 0.19%	+ 2.59%
LLaMA 7B	87.39 ± 0.081	87.98 ± 0.75	94.17 ± 0.32
vs. frozen	- 0.24%	+ 0.43%	+ 2.44%
LLaMA 65B	88.93 ± 0.056	**90.12** ± 0.51	**95.62** ± 0.17
vs. frozen	- 1.79%	+ 0.062%	+ 1.16%

The contrastively fine-tuned retriever was trained for one epoch to avoid overfitting, using three times as many negative pairs as positive pairs (roughly 5-10 mins depending on the dataset).

We note large improvements in the pure 1-NN mode accuracy, as expected, as we are optimizing a metric that is directly correlated with 1-NN performance. With fine-tuning, the pure 1-NN setup becomes near-competitive with ConvFit, the previous SoTA. In terms of retrieval+ICL performance, we see mixed results. In general the performance delta is quite small, suggesting that there is no significant retrieval quality bottleneck. In general, the fine-tuned CLINC retriever provides the most

boost, which is also the least data-scarce scenario (it is reasonable to expect the retriever fine-tuning to be more effective with more data).

3.13 RLHF models compared to unaligned LMs

Table 3.11: Comparison vs. RLHF-LLaMA (50-ex)

Model	**BANKING**	**HWU**
	10-shot	**10-shot**
LLaMA-2-7B	89.64	89.13
LLaMA-2-7B-chat	85.94	86.80

Looking at Table 3.11, we see that the RLHF-aligned LLM performs significantly worse at both BANKING and HWU than the base version, supporting the idea of a non-insignificant "alignment tax", mentioned in the seminal RLHF paper [33].

Chapter 4

In-Context Instruction Following

In the previous chapter we demonstrated that neural retrieval allows us to apply LLMs to tasks where the number of classes precludes us from using a fixed prompt of static demonstrations. Not only does retrieval enable us to use LLMs for these tasks, but LLMs are able to achieve SOTA results in such a setup. The next question we seek to tackle is: is instruction-following fine-tuning necessary to enable LLMs to follow instructions, or is it possible to use a similar retrieval-augmented regime to unlock this ability in-context? In the process, we will gather some evidence as to the "Superficial alignment hypothesis" mentioned in Chapter 2. If we are able to learn instruction following in-context, this suggests that fine-tuning for instruction following does not teach the model to perform any tasks that it hadn't already acquired the knowledge to be able to do from its pre-training.

4.1 Introduction

Most research on instruction-following thus far has focused on enabling LLMs to follow instructions via gradient-based optimization, i.e. supervised fine-tuning on description/input/output triples. Fine-tuning on large amounts of data in this format allows us to teach models to generalize to novel tasks, as the model learns *how to follow instructions*, rather than how to perform a specific task at training time. Supervised instruction-tuning is often accompanied by reinforcement learning from

human feedback (RLHF) to perform "alignment", or to bring model outputs in line with human expectations when interacting with the system. In this chapter, we focus on instruction tuning only (in a sense, a component or form of alignment itself), and see if we can replicate the results of instruction-tuning with supervised training by simply using ICL.

In this chapter, we focus on the following set of research questions:

1. Are we able to unlock instruction following capabilities in base language models without any fine-tuning using in-context learning with unrelated task demonstrations?
2. Does retrieval boost performance over using random demonstrations?
3. Is the base model able to generalize effectively to both classification and generation tasks?
4. How can we best condition the retrieval?
5. How closely does the ROUGE-L metric correlate with evaluation from a large SoTA model?
6. Is the base model able to generalize effectively to both classification and generation tasks?
7. Does the demonstration set matter?
8. Do unrelated task demonstrations help already instruction-tuned or RLHF-tuned models?

4.2 Method

We assume we have access to N instances of `(task description, input, output)` triples that compose our pool of demonstrations S. These demonstrations are for tasks that are completely unrelated to our current query/task. The goal of the work is to see if these unrelated demonstrations can nevertheless teach the model in-context how to follow instructions, such that it can complete the novel task it is seeing at test time. The intuition here is that the model may be able to learn the relationship between task description, input, and output through ICL, and thus be able to generalize effectively to the novel task.

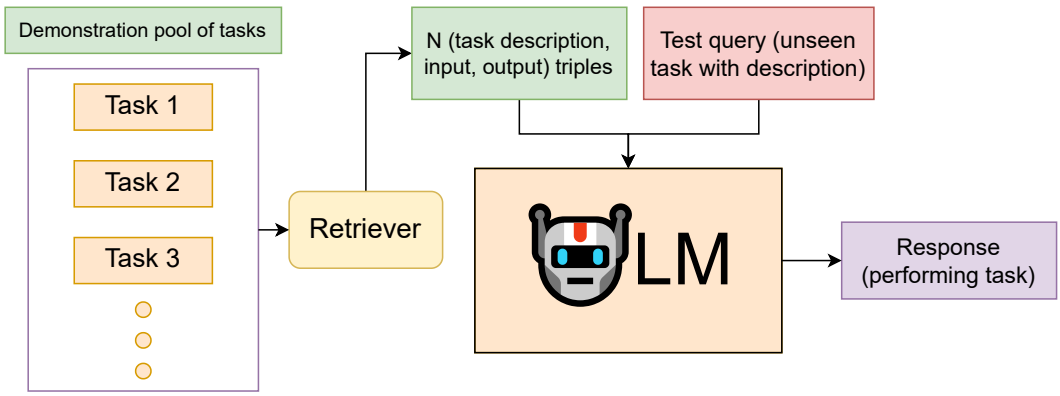

Figure 4.1: Complete pipeline for ICL Instruction Following

When retrieving from this pool of triples, we investigate 2 different ways we can query the retriever:

1. Querying based on the task description only
2. Querying jointly with task description and input

Alternatively, we can sample randomly from our pool of demonstrations.

When retrieving based on task description only, since each task will potentially have many data points, we retrieve randomly from within this set if we need to.

As with Chapter 3, we use a small dense retrieval model for retrieving the appropriate demonstrations.

4.3 Dataset

In our experimentation we use the existing dataset of Super-NaturalInstructions, which is composed of 1616 distinct natural language tasks. These tasks include both generative tasks, which may involve providing one word, a few words, or generating short free-form text; and classification tasks, which involve outputting a single or a few tokens representing a class name from a fixed set of classes. We take the first 100 instances from each of the 119 test tasks, leading to a test dataset of 11900 instances. This constitutes the official minimal test set, established by the authors of the paper. Some datapoint samples are provided in Table 4.1.

Table 4.1: Sample datapoints from SNI. Note that the third example contains a dataset error, the label should be "Past".

Instruction	Input	Output
In this task, you are given two questions about a domain. Your task is to combine the main subjects of the questions to write a new, natural-sounding question. (abriged)	What college did this president attend? Where did this president meet his wife?	Did this president meet his wife in college?
You are provided with an arithmetic question. Your task is to compute the solution using the given arithmetic operations. The only arithmetic operators needed to answer the questions are'+'(addition) and'-'(subtraction). The answer should be correct to one decimal place.	Tom has 9 yellow balloons Sara has 8 yellow balloons. How many yellow balloons do they have in total?	17.0
In this task you are given a sentence. You must judge whether the main verb of the sentence is in present or past tense. Label the instances as "Present" or "Past" based on your judgment. If there is no verb in the given text, answer "Present".	I quietly snuck up to him and pulled at his sleeve.	Present

4.4 Metrics

As mentioned previously, the tasks contained within Super-NaturalInstructions can broadly be split into two categories: generative and classification tasks. Generative tasks can be evaluated via ROUGE-L overlap, however it is well known that token overlap metrics are a poor proxy for actual generation quality, and can sometimes unfairly penalize models for reasonable generations, and thus need to be accompanied by additional metrics [1, 22]. This is especially the case when the set of ground-truth responses is relatively small, and thus does not adequately cover the space of all possible correct responses to the query (**this seems to be the case in SuperNI, where a majority of queries have only one ground-truth response**). As such, we provide a mix of several different metrics and testing setups. These additional metrics also seek to answer the research question of *which* kinds of tasks the model is able to learn to do in-context with unrelated demonstrations.

4.4.1 ROUGE-L

One of the metrics we provide is the standard ROUGE-L metric ordinarily used with the Super-NaturalInstructions dataset. We provide results using ROUGE-L on only the generative subset of the dataset, as the meaning of ROUGE-L on the classification subset is somewhat unclear. ROUGE-based metrics have some flaws (discussed extensively in Chapter 2), and thus we compliment the ROUGE metrics with several other metrics in our analysis.

4.4.2 Recall

A naive metric for the classification subset is exact match accuracy (EM). Unfortunately, through empirical testing we determined that EM vastly underestimates the performance of the LM. The reason for this is that these models will often provide additional textual content around the class name, instead of just the class name directly, which the exact match accuracy penalizes them for. For example, instead of returning the text "contradiction" for an entailment task, the LM may return "The class for this datapoint is contradiction. The explanation for this is...". As such, using a recall-based metric is more reasonable. In this case, the metric is checking if the correct class name is contained within the text the LM returns. This is not without issue however, as the LM's response may contain multiple class names, e.g. if it returns something along the lines of "I am unable to determine if this datapoint belongs to the contradiction or entailment classes". In this case, recall would score the response as correct, despite the model not actually making a choice.

4.4.3 Dense similarity on output

Another way to get around the issue with exact match accuracy is to perform an operation similar to what was used in Chapter 3, where as an additional post-processing step, we take the output of the language model and pass it through the dense similarity model we use for retrieval. As such, we constrain the pipeline's output to be one of the known classes for the given task, allowing the model some flexibility in returning the exact correct class name. For example, if the model were to return

"Sentence 1 entails sentence 2", then the dense similarity model would match this with the class name "entailment", despite the model not using the class name directly.

4.4.4 Evaluation from a large SoTA model

The final approach to evaluation is evaluation from a large SoTA model. In this case, we take the output of the model, and provide it to an already instruction-tuned language model, asking it to make a determination on whether the model's response successfully accomplished the task. There are two possible variants of the evaluation from a large SoTA model: reference-based and reference-free. In the reference-based case, we provide the evaluator model a reference answer in addition to the model response. This reference answer will be taken from the set of reference answers provided by the dataset and used to calculate the standard ROUGE-L score. In the reference-free case, the evaluator model is given the model response, and will have to make its own determination, with no grounding in any sample answers, of whether or not the model effectively solved the task with its response. In our case, since we have reference answers from the SuperNI dataset, we report only the grounded task completion rate.

4.5 Retrieval Sets

We evaluate the ability of LLaMA-7B to perform instruction-following via ICL on two different sets of demonstrations.

4.5.1 Super-NI itself

We use the train split of Super-NI itself as our first set of demonstrations. Using the train split ensures there is no task overlap between the demonstrations we will be providing to the model and the tasks we are evaluating it on. Our retrieval set is composed of the first 100 examples of each of the training tasks. This constitutes 75.600 triples examples in total.

4.5.2 Self-instruct Seed Task Set

Our second set of demonstrations comes from the Self-instruct paper [50], and is composed of the 175-example seed set of handwritten task-description-input-output examples that the authors use to build their instruction-tuning dataset. We use this as our set of demonstrations to see how little data we can make use of and still have an effective ICL-based instruction-following model.

4.6 Baselines

We provide 2 baselines/points of comparison for our ICL-based instruction-following model.

4.6.1 LLaMA zero-shot

Our first baseline is simply using the LLaMA model in zero-shot fashion, with no task demonstrations provided at all. We simply provide the query task description, input, and query the model for the output.

4.6.2 LLaMA-2-chat

With the release of LLaMA-2, there now exist open-access fully-RLHF-tuned models released by Meta. LLaMA-2-chat has been RLHF-tuned using human preference data, and thus fulfills both the criterion of instruction-following and also the criterion of broader harmlessness and helpfulness-based alignment. Given the use of external preference data, we expect that the LLaMA-2-chat model will likely perform better than our approach, which only uses Super-NI or the seed set of Self-Instruct.

Table 4.2: Performance of In-Context Instruction Following vs. Fully fine-tuned instruction-tuned models. "R-L" denotes the ROUGE-L score on the generative subset, "R" denotes the Recall score on the classification subset, and "D" denotes the EM score after the "dense similarity on model output" procedure. "L1" refers to LLaMA-1 and "L2" refers to LLaMA-2. "Saturated" refers to greedily filling the context window with examples. "Inst-only" refers to using just the instruction text to condition the retrieval; "joint" uses both the instruction text and the input text. If the model output mentions multiple classes in the classification task subset, we zero the prediction to not skew the recall numbers. "Random class selection" refers to selecting from the task classes randomly (classification subset only).

Model	**SuperNI Retrieval set**			**Self-Instruct Retrieval set**		
	R-L (gen)	**R (class)**	**D (class)**	**R-L (gen)**	**R (class)**	**D (class)**
Copying input	21.97	-	-	21.97	-	-
Random classes	-	43.29	43.29	-	43.29	43.29
L1 1-shot random	30.47 $\pm$ 9.00	43.76 $\pm$ 6.75	48.31 $\pm$ 1.23	31.37 $\pm$ 7.04	39.27 $\pm$ 6.33	47.83 $\pm$ 1.27
L1 Saturated random	37.59 $\pm$ 0.87	43.22 $\pm$ 0.80	49.07 $\pm$ 0.53	37.19 $\pm$ 0.77	41.67 $\pm$ 2.08	49.47 $\pm$ 0.40
L1 1-shot inst-only	36.07	41.29	48.45	35.29	40.45	48.29
L1 Saturated inst-only	35.79	41.73	47.63	**38.42**	41.39	48.55
L1 1-shot joint	35.52	41.94	47.86	32.51	40.49	47.69
L1 Saturated joint	**36.93**	**43.49**	**49.67**	36.32	**42.35**	**48.92**
L2 1-shot inst-only	37.86	42.37	49.31	34.50	40.84	48.27
L2 Saturated inst-only	37.53	42.67	49.55	**37.84**	**44.05**	**49.82**
L2 1-shot joint	37.22	42.92	**49.96**	33.77	40.14	48.10
L2 Saturated joint	**38.38**	**43.98**	49.43	35.88	43.90	49.16
L1 Zero-shot 7B	23.67	39.06	47.25	23.67	39.06	47.25
L2 Zero-shot 7B	26.34	**46.59**	48.94	26.34	**46.59**	48.94
L2 Z-S 7B-chat	**31.17**	42.84	**50.18**	**31.17**	42.84	**50.18**

4.7 Results and Discussion

4.7.1 Can we achieve zero-shot task generalization from a base LLM, with unrelated task demonstrations through ICL?

The answer to this research question seems to be yes, to some extent. When comparing to a strong RLHF-baseline (LLaMA-2-7B-chat), the prompted models are able to perform significantly better

on average on the SuperNI testing task subset using the ROUGE-L metrics (see Table 4.2, top performance from prompted model of **38.42** on the generative subset vs. **31.17** from the RLHF-tuned LLaMA-2-7B model). When using the alternative evaluation from a large SoTA model however, it is clear there is still a gap between the RLHF-tuned model and the best prompted model (see Table 4.5, LLaMA-2-7B-chat task completion rate of **34.9%** vs. best prompted model of **32.8%**). The gap is relatively small, and the improvement compared to just using the base model out-of-the-box zero-shot is stark (from Table 4.5, task completion rate of **22.4%** or **25.9%** for LLaMA-1 and LLaMA-2 respectively). Looking at Table 4.4, we also see that more examples generally correlates to better performance on the unrelated test-time tasks (on the generative subset only; on the classification subset, retrieval does not seem to help).

One interesting observation is that the performance of almost all models by default on the classification subset is below random chance (see Table 4.2, row "Random classes", **43.29**). This is surprising, as the expectation was that the classification tasks are the easier subset of the dataset, but this seems to not be the case. By using the dense similarity on output technique, all of the models climb to above random chance, but even the best performing model is only able to perform 6.89% better than random chance (see Table 4.2, LLaMA-2-chat performs at **50.18%** accuracy).

4.7.2 Does retrieval boost performance over random demonstration selection?

The answer to this research question seems to be somewhat mixed. In certain cases, retrieval does seem to boost performance compared to random selection, but the difference is relatively small (see Table 4.2). As such, retrieval does not seem to be an integral part of the instruction-following-through-ICL recipe, although it provides a bit of extra performance. Another argument for retrieval is that it is able to consistently give good results, while with random selection certain prompts provide poorer results and others better (see standard deviations for random demonstration selection in Table 4.2).

Worse performance with increasing examples on classification subset For the classification subset specifically, more examples seem to hurt model performance, especially if we are providing only classification examples in-context. Looking at Table 4.3, we see that the best performing models are all 1-shot (either instruction-only or joint conditioning).

One idea that we examined about this phenomenon was that overlapping class names between the demonstrations and the test task were causing the LM to become confused. As such, an experiment was run with avoiding overlapping class names between the demonstrations and test task (shown in Table 4.3, the "No-OV" rows). Removing the overlapping class names improved the performance very slightly (regular score of **48.55%** on the classification subset, adding No-OV improves the score to **48.83%** for the instruction-only conditioning case).

4.7.3 Is the base model able to generalize effectively to both classification and generation tasks?

It seems to be the case that there is a difference between the classification and generative subsets of the dataset. Specifically, it seems that the 7B base-model, even with no prompting or tuning, and equipped with the dense similarity on output, is able to perform meaningfully better than the random class selection baseline (see Table 4.2, LLaMA-1 zero-shot base model gets **47.25%**, LLaMA-2 zero-shot base model gets **48.94%**, random chance is **43.29%**). However none of the 7B models are able to significantly improve beyond this random selection baseline (the best improvement is 6.89% above the baseline, with **50.18%**). This may imply that not only are these specific classification tasks difficult to generalize to in-context, but also that none of the models were able to generalize effectively from their pre-training data to be able to solve these classification tasks. This also seems to support the Superficial Alignment hypothesis, in that even the RLHF-tuned model was unable to perform well in them. In other words, even instruction fine-tuning and human-preference-based-tuning was unable to teach the 7B-chat model to solve these tasks, supporting the idea that alignment tuning is unable to teach the model to perform new tasks it has not already seen in some form during pre-training.

In terms of the types of demonstrations we provide, we can see in Table 4.3 that providing classification-only demonstrations slightly boosts performance on the classification set, but not on the generative split (peak score with LLaMA-1 and all examples, generative and classification, of **48.92%** on the classification split from Table 4.2, while giving class-only examples gives a performance of **49.98%** with the same base model). However, given that this method is simply prompting, one can envision a real-world use case where a different prompt (set of demonstrations) is given to the model depending on if the user's query is a classification or generative task.

Table 4.3: Ablation with classification-only demonstrations, using the Self-Instruct Retrieval set. "No-OV" refers to not selecting demonstrations whose class names overlap with the class names of the task the query belongs to.

Model	**R-L (gen)**	**R (class)**	**D (class)**
1-shot class-only random	35.05 ± 0.69	42.95 ± 0.65	49.00 ± 0.79
5-shot class-only random	35.68 ± 0.53	43.73 ± 0.54	48.97 ± 0.48
1-shot class-only inst-only	34.64	41.16	**49.20**
5-shot class-only inst-only	34.46	41.84	46.29
10-shot class-only inst-only	34.54	42.57	47.33
Saturated class-only inst-only	34.74	41.59	46.88
Saturated inst-only No-OV	38.49	40.61	48.63
Saturated inst-only class-only No-OV	35.41	41.57	46.57
1-shot class-only joint	34.66	43.76	**49.98**
5-shot class-only joint	34.94	42.05	48.02
10-shot class-only joint	35.02	41.00	46.43
Saturated class-only joint	35.33	41.61	46.53
Saturated joint No-OV	35.65	42.82	48.94
Saturated joint class-only No-OV	35.41	41.57	46.57

4.7.4 How can we best condition the retrieval?

The best way to condition the retrieval seems to be somewhat dependent on the demonstration dataset and type of test-time query. Using the Self-Instruct Retrieval set, the instruction-only retrieval mostly wins out (Table 4.2, in both the LLaMA-1 and LLaMA-2 categories instruction-only wins against joint conditioning for the generative subset, with **38.42** and **37.84** ROUGE-L scores respectively vs.

the joint scores of **36.32** and **35.88** respectively). However, for the classification subset, it seems that in certain cases the joint conditioning also wins. In general, as mentioned previously, retrieval seems to mostly hurt performance on the classification split.

Table 4.4: Ablations across number of demonstrations, using the Self-Instruct Retrieval set (LLaMA-1).

Model	R-L (gen)	R (class)	D (class)
1-shot inst-only	35.29	40.45	48.29
5-shot inst-only	37.27	42.22	**49.00**
10-shot inst-only	38.14	43.04	48.75
15-shot inst-only	38.38	41.22	48.41
Saturated inst-only	**38.42**	41.39	48.55
1-shot joint	32.51	40.49	47.69
5-shot joint	35.57	41.74	**49.31**
10-shot joint	35.95	42.16	49.11
15-shot joint	36.05	42.78	49.00
Saturated joint	**36.32**	42.35	48.92

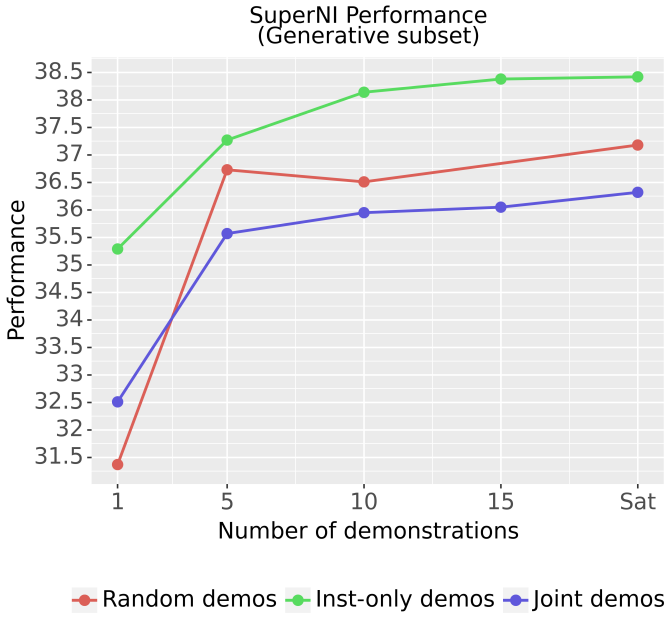

Figure 4.2: Performance vs. Number of Examples on SuperNI test subset

Table 4.5: Performance of In-Context Instruction Following vs. Fully fine-tuned instruction-tuned models, evaluation from a large SoTA model (GPT-4-Turbo). Numbers are from the generative subset only (sample of 1000).

Model	Task Completion Success Rate (GPT-4-Turbo)
Zero-shot LLaMA-1-7B	22.4%
Zero-shot LLaMA-2-7B	25.9%
1-shot ICL LLaMA-1-7B (inst)	28.0%
Saturated ICL LLaMA-2-7B (joint, SNI retrieval)	30.5%
1-shot ICL LLaMA-2-7B (inst)	31.4%
Saturated ICL LLaMA-1-7B (inst)	31.9%
Saturated ICL LLaMA-2-7B (inst)	32.8%
LLaMA-2-7B-chat	34.9%
Sat ICL + LLaMA-2-7B-chat	**47.7%**

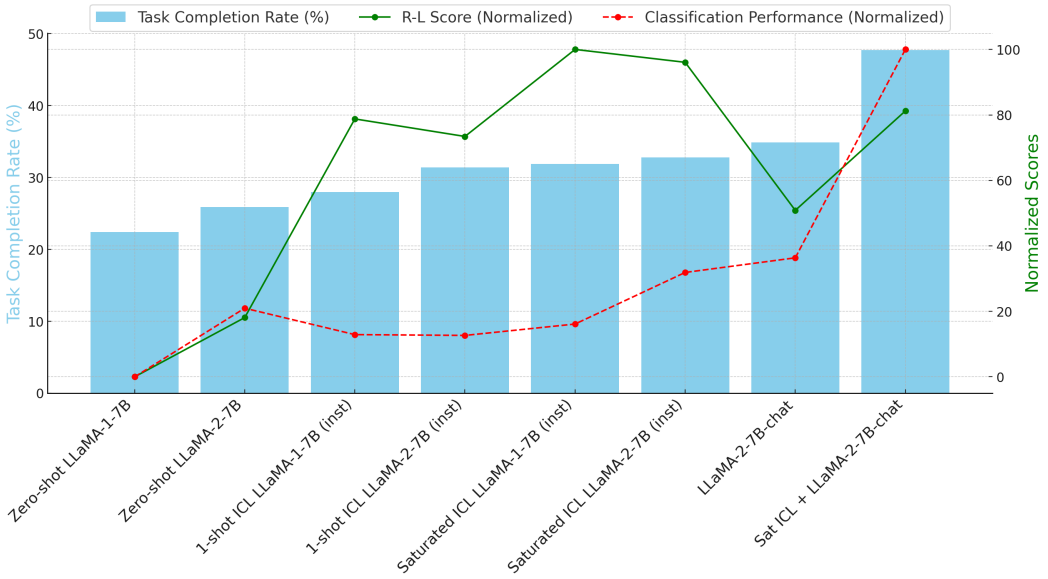

Figure 4.3: Comparison of ROUGE-L vs. LLM-based vs. classification evaluation. We can see that the performance of LLaMA-2-7B-chat seems to be underestimated by ROUGE-L significantly. The classification performance seems to be a more accurate predictor of the LLM-based evaluation result.

4.7.5 How closely does the ROUGE-L metric correlate with evaluation from a large SoTA model?

The evaluation from a large SoTA model seems to only partially correlate with ROUGE-L (see Figure 4.3 for a visual comparison of the two metrics). For example, with a single example in-context, we already beat LLaMA-2-7B-chat, the RLHF-tuned model, in terms of generative ROUGE-L (see Table 4.2 **35.29** ROUGE-L score for 1-shot vs. **31.17** chat). However, when comparing with the evaluation from the large SoTA model with task completion rate, we see that LLaMA-2-7B-chat actually wins, although the margin is reasonable (LLaMA-2-7B-chat with a completion rate of **34.9%** vs. the top prompted model completion rate of **32.8%**). The ROUGE-L scores seem to be overestimating the performance of the prompted models, and underestimating the performance of the 7B RLHF-tuned model (see visual comparison in Figure 4.3).

4.7.6 LLaMA-2-7B-Chat Failure Mode: Output Description

One failure mode observed in the LLaMA-2-7B-chat model is that the model will simply describe what the output should look like rather than providing an actual answer to the query. For example, for the task of writing a title for a given article, the model may respond with "A title for the article." rather than actually performing the task. It was observed that whenever this behaviour occurs, it occurs for every datapoint of a given task, regardless of what the input is, such that there are entire tasks that the Chat model is unable to perform and gets a 0 ROUGE-L score on.

4.7.7 Does the set of demonstrations matter?

One interesting observation is that the set of demonstrations does not seem to particularly matter in terms of triggering the base model's "instruction-following mode" in-context. Looking at Table 4.2, we see slight differences between using the SuperNI retrieval set vs. the Self-Instruct retrieval set (performance differences of <1%). However, it is worth noting that the Self-Instruct set is composed of 175 examples, while the SuperNI Retrieval set is composed of 75.600 examples. It

seems safe to conclude that the size of the demonstration set plays a very small role, if any, in the final performance.

Table 4.6: In-context learning with already fine-tuned models, using the Self-Instruct Retrieval set.

Model	R-L (gen)	R (class)	D (class)
LLaMA-2-7B-chat (unprompted)	31.17	42.84	50.18
LLaMA-2-7B-chat (prompted)	**35.65**	**45.29**	**55.31**

4.7.8 Do unrelated task demonstrations help already instruction-tuned or RLHF-tuned models?

Already-instruction-tuned models also seem to significantly benefit from unrelated task demonstrations. Looking at Table 4.6, we see a reasonably large boost in ROUGE-L on the generative subset as a result of using a similar prompting setup but with LLaMA-2-7B-chat which has been RLHF-tuned (**35.65** vs. **31.17** ROUGE-L score) . With the dense-similarity-on-output mechanism on the classification subset, we see the strongest classification performance out of all the models tested (**55.31%** accuracy). Although the generation ROUGE-L is also improved, it is not the strongest ROUGE-L out of all models tested, which rests with the Saturated instruction-only ICL base model. However, an interesting observation is that in the evaluation from a large SoTA model (Table 4.5), combining prompting with the already RLHF-tuned model provides the highest task completion success rate out of **all 7B models tested** (**47.7%** task completion success rate, a significant jump from the zero-shot LLaMA-2-chat of **34.9%**). This demonstrates that the irrelevant task prompting method is actually *complimentary* to RLHF tuning, and not necessarily in contrast with it. The implications of this are interesting; one conclusion to be drawn could be that existing methods for instruction-tuning/RLHF-tuning are not fully "capturing" the model's zero-shot task generalization abilities.

Chapter 5

Conclusion

5.1 Summary of Thesis

In this thesis, we investigated the use of retrieval systems in combination with in-context learning. We used dense retrieval models to enable the use of LLMs for classification tasks with large label sets, where ordinarily context window limitations would make the use of LLMs difficult. In addition, we investigate using ICL for instruction following, to avoid the regular procedure of regular instruction-tuning and RLHF alignment, which is extremely costly due to the degree of human input it requires.

5.2 Contributions to the Literature

Our contributions to the literature are several. We achieved SOTA performance with 4 different short text classification datasets, where the number of classes ranges from 28 to 150. We analyzed model performance across different model scales and numbers of demonstrations, showing that larger models are better able to use more demonstrations effectively. We investigated the effect of the ordering of demonstrations in-context, showing that the order of demonstrations matters significantly, and retrieval models can be used simply to reorder demonstrations for a performance boost. We performed several ablations on different aspects of the demonstrations, demonstrating that

the similarity of the query to the demonstrations, the correct input-output correspondence, and the semantically-significant class names all play a significant role in the final performance of the system. We compare with classical (BM25) and larger neural retrievers, showing that larger retrievers are not necessarily more performant. We also demonstrate that fine-tuning the retriever does not necessarily lead to performance gain, especially when data is the most scarce. The contribution of this part of the work concludes with showing that RLHF-aligned LLMs perform worse on the task than their base versions, supporting the idea of a non-insignificant "alignment tax", mentioned in the seminal RLHF paper [33].

In the second part of the thesis, we investigate the use of ICL for instruction-following, with the dual aim of a) reducing the costliness of training instruction-tuned and RLHF-aligned models, and b) investigating the "Superficial Alignment Hypothesis". We demonstrate that using irrelevant task demonstrations greatly boosts the instruction-following ability of the base model, however according to the model-based evaluation a gap still exists between the prompted and RLHF-tuned equivalent-model. However, we demonstrate that the irrelevant task demonstration approach is actually complimentary to RLHF-tuning, and improves the performance of the RLHF-tuned model both in ROUGE-L scores and also in model-based evaluation, where the prompted and RLHF-tuned model reached the highest performance out of all the 7B models tested.

5.3 Future Work

Future work extending retrieval-augmented in-context learning for complex classification tasks with large label sets (the content of Chapter 3) could involve scaling up the size of the label set even further, into hundreds or thousands of labels, to investigate if the LLM is still able to perform the task effectively in combination with the retrieval model. Beyond this, although all of the alternative retrieval strategies that we tried did not perform meaningfully better than nearest neighbor, it is possible that there are even more complex strategies that would. Future work could investigate other strategies, or perhaps investigate why exactly nearest neighbor is so effective. More specifically, we do not have a compelling explanation for why increasing the diversity of in-context examples

does not boost performance, as it intuitively seems like it should. The idea of removing "redundant examples" seems intuitive but fails in practice, and the mechanism behind why this happens is still very unclear.

Future work extending the use of in-context learning for instruction following (the contents of Chapter 4) could take the form of investigating the reason behind why the 7B-chat model's performance is so severely underestimated by the ROUGE-L metric. Future work could also investigate the use of in-context learning more broadly for alignment, rather than just for instruction following, as was done in this thesis. Instruction following is only one component of alignment; and since ICL has been demonstrated to perform well in this context, it's possible it can be used more broadly for safety and other types of alignment as well.

Bibliography

[1] AKTER, M., BANSAL, N., AND KARMAKER, S. K. Revisiting automatic evaluation of extractive summarization task: Can we do better than ROUGE? In *Findings of the Association for Computational Linguistics: ACL 2022* (Dublin, Ireland, May 2022), S. Muresan, P. Nakov, and A. Villavicencio, Eds., Association for Computational Linguistics, pp. 1547–1560.

[2] ASKELL, A., BAI, Y., CHEN, A., DRAIN, D., GANGULI, D., HENIGHAN, T., JONES, A., JOSEPH, N., MANN, B., DASSARMA, N., ELHAGE, N., HATFIELD-DODDS, Z., HERNANDEZ, D., KERNION, J., NDOUSSE, K., OLSSON, C., AMODEI, D., BROWN, T., CLARK, J., MCCANDLISH, S., OLAH, C., AND KAPLAN, J. A General Language Assistant as a Laboratory for Alignment, Dec. 2021. arXiv:2112.00861 [cs].

[3] BOMMASANI, R., HUDSON, D. A., ADELI, E., ALTMAN, R., ARORA, S., VON ARX, S., BERNSTEIN, M. S., BOHG, J., BOSSELUT, A., BRUNSKILL, E., BRYNJOLFSSON, E., BUCH, S., CARD, D., CASTELLON, R., CHATTERJI, N., CHEN, A., CREEL, K., DAVIS, J. Q., DEMSZKY, D., DONAHUE, C., DOUMBOUYA, M., DURMUS, E., ERMON, S., ETCHEMENDY, J., ETHAYARAJH, K., FEI-FEI, L., FINN, C., GALE, T., GILLESPIE, L., GOEL, K., GOODMAN, N., GROSSMAN, S., GUHA, N., HASHIMOTO, T., HENDERSON, P., HEWITT, J., HO, D. E., HONG, J., HSU, K., HUANG, J., ICARD, T., JAIN, S., JURAFSKY, D., KALLURI, P., KARAMCHETI, S., KEELING, G., KHANI, F., KHATTAB, O., KOH, P. W., KRASS, M., KRISHNA, R., KUDITIPUDI, R., KUMAR, A., LADHAK, F., LEE, M., LEE, T., LESKOVEC, J., LEVENT, I., LI, X. L., LI, X., MA, T., MALIK, A., MANNING, C. D., MIRCHANDANI, S., MITCHELL, E., MUNYIKWA, Z., NAIR, S., NARAYAN, A.,

NARAYANAN, D., NEWMAN, B., NIE, A., NIEBLES, J. C., NILFOROSHAN, H., NYARKO, J., OGUT, G., ORR, L., PAPADIMITRIOU, I., PARK, J. S., PIECH, C., PORTELANCE, E., POTTS, C., RAGHUNATHAN, A., REICH, R., REN, H., RONG, F., ROOHANI, Y., RUIZ, C., RYAN, J., RÉ, C., SADIGH, D., SAGAWA, S., SANTHANAM, K., SHIH, A., SRINIVASAN, K., TAMKIN, A., TAORI, R., THOMAS, A. W., TRAMÈR, F., WANG, R. E., WANG, W., WU, B., WU, J., WU, Y., XIE, S. M., YASUNAGA, M., YOU, J., ZAHARIA, M., ZHANG, M., ZHANG, T., ZHANG, X., ZHANG, Y., ZHENG, L., ZHOU, K., AND LIANG, P. On the opportunities and risks of foundation models, 2022.

[4] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 1877–1901.

[5] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems* (2020), vol. 33, Curran Associates, Inc., pp. 1877–1901.

[6] CASANUEVA, I., TEMCINAS, T., GERZ, D., HENDERSON, M., AND VULIC, I. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020* (mar 2020). Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

[7] CHAN, S. C. Y., SANTORO, A., LAMPINEN, A. K., WANG, J. X., SINGH, A., RICHEMOND, P. H., MCCLELLAND, J., AND HILL, F. Data distributional properties drive emergent in-context learning in transformers, 2022.

[8] CHIANG, W.-L., LI, Z., LIN, Z., SHENG, Y., WU, Z., ZHANG, H., ZHENG, L., ZHUANG, S., ZHUANG, Y., GONZALEZ, J. E., STOICA, I., AND XING, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[9] CHUNG, H. W., HOU, L., LONGPRE, S., ZOPH, B., TAY, Y., FEDUS, W., LI, Y., WANG, X., DEHGHANI, M., BRAHMA, S., WEBSON, A., GU, S. S., DAI, Z., SUZGUN, M., CHEN, X., CHOWDHERY, A., CASTRO-ROS, A., PELLAT, M., ROBINSON, K., VALTER, D., NARANG, S., MISHRA, G., YU, A., ZHAO, V., HUANG, Y., DAI, A., YU, H., PETROV, S., CHI, E. H., DEAN, J., DEVLIN, J., ROBERTS, A., ZHOU, D., LE, Q. V., AND WEI, J. Scaling instruction-finetuned language models, 2022.

[10] DEMSZKY, D., MOVSHOVITZ-ATTIAS, D., KO, J., COWEN, A., NEMADE, G., AND RAVI, S. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 4040–4054.

[11] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] DUBOIS, Y., LI, X., TAORI, R., ZHANG, T., GULRAJANI, I., BA, J., GUESTRIN, C., LIANG, P., AND HASHIMOTO, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

[13] EKMAN, P. An argument for basic emotions. *Cognition and Emotion 6*, 3-4 (1992), 169–200.

[14] GAO, T., FISCH, A., AND CHEN, D. 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) (Online, Aug. 2021), C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Association for Computational Linguistics, pp. 3816–3830.

[15] GARG, S., TSIPRAS, D., LIANG, P., AND VALIANT, G. What can transformers learn in-context? a case study of simple function classes, 2023.

[16] GLAESE, A., MCALEESE, N., TRĘBACZ, M., ASLANIDES, J., FIROIU, V., EWALDS, T., RAUH, M., WEIDINGER, L., CHADWICK, M., THACKER, P., CAMPBELL-GILLINGHAM, L., UESATO, J., HUANG, P.-S., COMANESCU, R., YANG, F., SEE, A., DATHATHRI, S., GREIG, R., CHEN, C., FRITZ, D., ELIAS, J. S., GREEN, R., MOKRÁ, S., FERNANDO, N., WU, B., FOLEY, R., YOUNG, S., GABRIEL, I., ISAAC, W., MELLOR, J., HASSABIS, D., KAVUKCUOGLU, K., HENDRICKS, L. A., AND IRVING, G. Improving alignment of dialogue agents via targeted human judgements, 2022.

[17] HOFFMANN, J., BORGEAUD, S., MENSCH, A., BUCHATSKAYA, E., CAI, T., RUTHERFORD, E., DE LAS CASAS, D., HENDRICKS, L. A., WELBL, J., CLARK, A., HENNIGAN, T., NOLAND, E., MILLICAN, K., VAN DEN DRIESSCHE, G., DAMOC, B., GUY, A., OSINDERO, S., SIMONYAN, K., ELSEN, E., RAE, J. W., VINYALS, O., AND SIFRE, L. Training compute-optimal large language models, 2022.

[18] HUPKES, D., GIULIANELLI, M., DANKERS, V., ARTETXE, M., ELAZAR, Y., PIMENTEL, T., CHRISTODOULOPOULOS, C., LASRI, K., SAPHRA, N., SINCLAIR, A., ULMER, D., SCHOTTMANN, F., BATSUREN, K., SUN, K., SINHA, K., KHALATBARI, L., RYSKINA, M., FRIESKE, R., COTTERELL, R., AND JIN, Z. State-of-the-art generalisation research in nlp: A taxonomy and review, 2023.

[19] KARPUKHIN, V., OGUZ, B., MIN, S., LEWIS, P., WU, L., EDUNOV, S., CHEN, D., AND YIH, W.-T. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 6769–6781.

[20] KARPUKHIN, V., OĞUZ, B., MIN, S., LEWIS, P., WU, L., EDUNOV, S., CHEN, D., AND TAU YIH, W. Dense passage retrieval for open-domain question answering, 2020.

[21] LARSON, S., MAHENDRAN, A., PEPER, J. J., CLARKE, C., LEE, A., HILL, P., KUMMERFELD, J. K., LEACH, K., LAURENZANO, M. A., TANG, L., AND MARS, J. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 1311–1316.

[22] LIANG, P., BOMMASANI, R., LEE, T., TSIPRAS, D., SOYLU, D., YASUNAGA, M., ZHANG, Y., NARAYANAN, D., WU, Y., KUMAR, A., NEWMAN, B., YUAN, B., YAN, B., ZHANG, C., COSGROVE, C., MANNING, C. D., RÉ, C., ACOSTA-NAVAS, D., HUDSON, D. A., ZELIKMAN, E., DURMUS, E., LADHAK, F., RONG, F., REN, H., YAO, H., WANG, J., SANTHANAM, K., ORR, L., ZHENG, L., YUKSEKGONUL, M., SUZGUN, M., KIM, N., GUHA, N., CHATTERJI, N., KHATTAB, O., HENDERSON, P., HUANG, Q., CHI, R., XIE, S. M., SANTURKAR, S., GANGULI, S., HASHIMOTO, T., ICARD, T., ZHANG, T., CHAUDHARY, V., WANG, W., LI, X., MAI, Y., ZHANG, Y., AND KOREEDA, Y. Holistic evaluation of language models, 2023.

[23] LIU, J., SHEN, D., ZHANG, Y., DOLAN, B., CARIN, L., AND CHEN, W. 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* (Dublin, Ireland and Online, May 2022), Association for Computational Linguistics, pp. 100–114.

[24] LIU, N. F., LIN, K., HEWITT, J., PARANJAPE, A., BEVILACQUA, M., PETRONI, F., AND LIANG, P. Lost in the middle: How language models use long contexts, 2023.

[25] LIU, X., ESHGHI, A., SWIETOJANSKI, P., AND RIESER, V. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*. Springer Singapore, Singapore, 2021, pp. 165–183.

[26] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTLEMOYER, L., AND STOYANOV, V. Roberta: A robustly optimized bert pretraining approach, 2019.

[27] LU, Y., BARTOLO, M., MOORE, A., RIEDEL, S., AND STENETORP, P. 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)* (Dublin, Ireland, May 2022), Association for Computational Linguistics, pp. 8086–8098.

[28] MEHRI, S., ERIC, M., AND HAKKANI-TÜR, D. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR abs/2009.13570* (2020).

[29] MILIOS, A., REDDY, S., AND BAHDANAU, D. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP* (Singapore, Dec. 2023), D. Hupkes, V. Dankers, K. Batsuren, K. Sinha, A. Kazemnejad, C. Christodoulopoulos, R. Cotterell, and E. Bruni, Eds., Association for Computational Linguistics, pp. 173–184.

[30] MIN, S., LYU, X., HOLTZMAN, A., ARTETXE, M., LEWIS, M., HAJISHIRZI, H., AND ZETTLEMOYER, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Abu Dhabi, United Arab Emirates, Dec. 2022), Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Association for Computational Linguistics, pp. 11048–11064.

[31] MIN, S., LYU, X., HOLTZMAN, A., ARTETXE, M., LEWIS, M., HAJISHIRZI, H., AND ZETTLEMOYER, L. Rethinking the role of demonstrations: What makes in-context learning work?, 2022.

[32] MISHRA, S., KHASHABI, D., BARAL, C., AND HAJISHIRZI, H. Cross-task generalization via natural language crowdsourcing instructions, 2022.

[33] OUYANG, L., WU, J., JIANG, X., ALMEIDA, D., WAINWRIGHT, C., MISHKIN, P., ZHANG, C., AGARWAL, S., SLAMA, K., RAY, A., SCHULMAN, J., HILTON, J., KELTON, F., MILLER, L., SIMENS, M., ASKELL, A., WELINDER, P., CHRISTIANO, P. F., LEIKE, J., AND LOWE, R. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems 35* (Dec. 2022), 27730–27744.

[34] OUYANG, L., WU, J., JIANG, X., ALMEIDA, D., WAINWRIGHT, C. L., MISHKIN, P., ZHANG, C., AGARWAL, S., SLAMA, K., RAY, A., SCHULMAN, J., HILTON, J., KELTON, F., MILLER, L., SIMENS, M., ASKELL, A., WELINDER, P., CHRISTIANO, P., LEIKE, J., AND LOWE, R. Training language models to follow instructions with human feedback, 2022.

[35] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[36] RAM, O., LEVINE, Y., DALMEDIGOS, I., MUHLGAY, D., SHASHUA, A., LEYTON-BROWN, K., AND SHOHAM, Y. In-context retrieval-augmented language models, 2023.

[37] RAZEGHI, Y., AU2, R. L. L. I., GARDNER, M., AND SINGH, S. Impact of pretraining term frequencies on few-shot reasoning, 2022.

[38] RAZEGHI, Y., LOGAN IV, R. L., GARDNER, M., AND SINGH, S. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022* (Abu Dhabi, United Arab Emirates, Dec. 2022), Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Association for Computational Linguistics, pp. 840–854.

[39] REIMERS, N., AND GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[40] REIMERS, N., AND GUREVYCH, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3982–3992.

[41] REIMERS, N., AND GUREVYCH, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3982–3992.

[42] REYNOLDS, L., AND MCDONELL, K. Prompt programming for large language models: Beyond the few-shot paradigm, 2021.

[43] ROBERTSON, S. E., WALKER, S., JONES, S., HANCOCK-BEAULIEU, M., AND GATFORD, M. Okapi at trec-3. In *Text Retrieval Conference* (1994).

[44] SAHU, G., RODRIGUEZ, P., LARADJI, I., ATIGHEHCHIAN, P., VAZQUEZ, D., AND BAHDANAU, D. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI* (Dublin, Ireland, May 2022), B. Liu, A. Papangelis, S. Ultes, A. Rastogi, Y.-N. Chen, G. Spithourakis, E. Nouri, and W. Shi, Eds., Association for Computational Linguistics, pp. 47–57.

[45] SHI, W., MIN, S., YASUNAGA, M., SEO, M., JAMES, R., LEWIS, M., ZETTLEMOYER, L., AND TAU YIH, W. Replug: Retrieval-augmented black-box language models, 2023.

[46] SPARCK JONES, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*, 1 (1972), 11–21.

[47] TOUVRON, H., LAVRIL, T., IZACARD, G., MARTINET, X., LACHAUX, M.-A., LACROIX, T., ROZIÈRE, B., GOYAL, N., HAMBRO, E., AZHAR, F., RODRIGUEZ, A., JOULIN, A., GRAVE, E., AND LAMPLE, G. Llama: Open and efficient foundation language models, 2023.

[48] TOUVRON, H., MARTIN, L., STONE, K., ALBERT, P., ALMAHAIRI, A., BABAEI, Y., BASHLYKOV, N., BATRA, S., BHARGAVA, P., BHOSALE, S., BIKEL, D., BLECHER, L., FERRER, C. C., CHEN, M., CUCURULL, G., ESIOBU, D., FERNANDES, J., FU, J., FU, W., FULLER, B., GAO, C., GOSWAMI, V., GOYAL, N., HARTSHORN, A., HOSSEINI, S., HOU, R., INAN, H., KARDAS, M., KERKEZ, V., KHABSA, M., KLOUMANN, I., KORENEV, A., KOURA, P. S., LACHAUX, M.-A., LAVRIL, T., LEE, J., LISKOVICH, D., LU, Y., MAO, Y., MARTINET, X., MIHAYLOV, T., MISHRA, P., MOLYBOG, I., NIE, Y., POULTON, A., REIZENSTEIN, J., RUNGTA, R., SALADI, K., SCHELTEN, A., SILVA, R., SMITH, E. M., SUBRAMANIAN, R., TAN, X. E., TANG, B., TAYLOR, R., WILLIAMS, A., KUAN, J. X., XU, P., YAN, Z., ZAROV, I., ZHANG, Y., FAN, A., KAMBADUR, M., NARANG, S., RODRIGUEZ, A., STOJNIC, R., EDUNOV, S., AND SCIALOM, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

[49] VULIĆ, I., SU, P.-H., COOPE, S., GERZ, D., BUDZIANOWSKI, P., CASANUEVA, I., MRKŠIĆ, N., AND WEN, T.-H. ConvFiT: Conversational fine-tuning of pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Online and Punta Cana, Dominican Republic, Nov. 2021), Association for Computational Linguistics, pp. 1151–1168.

[50] WANG, Y., KORDI, Y., MISHRA, S., LIU, A., SMITH, N. A., KHASHABI, D., AND HAJISHIRZI, H. Self-Instruct: Aligning Language Model with Self Generated Instructions, Dec. 2022. arXiv:2212.10560 [cs].

[51] WANG, Y., MISHRA, S., ALIPOORMOLABASHI, P., KORDI, Y., MIRZAEI, A., ARUNKUMAR, A., ASHOK, A., DHANASEKARAN, A. S., NAIK, A., STAP, D., PATHAK, E., KARAMANOLAKIS, G., LAI, H. G., PUROHIT, I., MONDAL, I., ANDERSON, J., KUZNIA, K., DOSHI, K., PATEL, M., PAL, K. K., MORADSHAHI, M., PARMAR, M., PUROHIT, M., VARSHNEY, N., KAZA, P. R., VERMA, P., PURI, R. S., KARIA, R., SAMPAT, S. K., DOSHI, S., MISHRA, S., REDDY, S., PATRO, S., DIXIT, T., SHEN, X., BARAL, C., CHOI,

Y., SMITH, N. A., HAJISHIRZI, H., AND KHASHABI, D. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

[52] WEI, J., WEI, J., TAY, Y., TRAN, D., WEBSON, A., LU, Y., CHEN, X., LIU, H., HUANG, D., ZHOU, D., AND MA, T. Larger language models do in-context learning differently, 2023.

[53] XIE, S. M., RAGHUNATHAN, A., LIANG, P., AND MA, T. An explanation of in-context learning as implicit bayesian inference, 2022.

[54] ZHANG, S., ROLLER, S., GOYAL, N., ARTETXE, M., CHEN, M., CHEN, S., DEWAN, C., DIAB, M., LI, X., LIN, X. V., MIHAYLOV, T., OTT, M., SHLEIFER, S., SHUSTER, K., SIMIG, D., KOURA, P. S., SRIDHAR, A., WANG, T., AND ZETTLEMOYER, L. Opt: Open pre-trained transformer language models, 2022.

[55] ZHAO, T. Z., WALLACE, E., FENG, S., KLEIN, D., AND SINGH, S. Calibrate before use: Improving few-shot performance of language models, 2021.

[56] ZHOU, C., LIU, P., XU, P., IYER, S., SUN, J., MAO, Y., MA, X., EFRAT, A., YU, P., YU, L., ZHANG, S., GHOSH, G., LEWIS, M., ZETTLEMOYER, L., AND LEVY, O. Lima: Less is more for alignment, 2023.