

# Making LLMs Worth Every Penny: Resource-Limited Text Classification in Finance

Lefteris Loukas  
lefteris.loukas@helvia.ai  
Helvia.ai  
Department of Informatics, Athens  
University of Economics and Business  
Greece

Ilias Stogiannidis  
ilias.stogiannidis@helvia.ai  
Helvia.ai  
Department of Informatics, Athens  
University of Economics and Business  
Greece

Odysseas Diamantopoulos  
odysseas.diamantopoulos@helvia.ai  
Helvia.ai  
Department of Informatics, Athens  
University of Economics and Business  
Greece

Prodromos Malakasiotis  
Department of Informatics, Athens  
University of Economics and Business  
Greece

Stavros Vassos  
stavros@helvia.ai  
Helvia.ai  
Greece

## ABSTRACT

Intent detection can be tackled in several ways in NLP. Standard Full-Data classifiers demand thousands of labeled examples, which is impractical in data-limited domains. Few-shot methods offer an alternative, utilizing modern contrastive learning techniques that can be effective with as little as 20 examples per class. Similarly, generative Large Language Models (LLMs) like GPT-4 can perform effectively with just 1-5 examples per class. However, the trade-offs between performance and costs for these approaches remain under-explored, a critical concern for budget-limited organizations. Our work addresses this gap by offering a study of the aforementioned approaches over the Banking77 financial intent detection dataset, including the evaluation of cutting-edge LLMs by OpenAI, Cohere, and Anthropic in a comprehensive set of few-shot scenarios. We complete the picture with two additional methods: first, a cost-effective querying method for LLMs based on retrieval-augmented generation, which is able to reduce operational costs up to 22 times compared to classic few-shot approaches, and second, a data augmentation method using GPT-4, which is able to improve performance and offers benefits in resource-limited scenarios. Finally, to inspire future research, we provide a human expert's curated subset of Banking77, along with extensive error analysis.

## CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence; Natural language processing; Information extraction;**

## KEYWORDS

Financial, NLP Applications, LLMs, OpenAI, GPT, Anthropic, Claude, Cohere, Banking77, SetFit, Few-Shot, Intent Classification

## ACM Reference Format:

Lefteris Loukas, Ilias Stogiannidis, Odysseas Diamantopoulos, Prodromos Malakasiotis, and Stavros Vassos. XYZ2. Making LLMs Worth Every Penny: Resource-Limited Text Classification in Finance. In *Proceedings of 4th ACM International Conference on AI in Finance (ICAIF'23)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

## 1 INTRODUCTION

The field of Natural Language Processing (NLP) has seen impressive advancements in the past few years, with particular emphasis on text classification. Traditional full-data classifiers require thousands of labeled samples, making them infeasible for data-limited domains such as finance [5]. Modern Few-Shot techniques, which include contrastive learning [40], aim to alleviate this issue by requiring only 10 to 20 examples per class. Also, recent advancements focus on prompting large Language Models (LLMs) like GPT-3 with as few as 1-5 examples per class, typically via a managed API. However, the tradeoffs concerning the performance and Operating Costs (OpEx) of available methods remain under-explored.

In this paper, we bridge this gap by evaluating the aforementioned approaches in a comprehensive set of few-shot scenarios over a financial intent classification dataset, Banking77 [5], including the evaluation of cutting-edge LLMs by OpenAI, Cohere, and Anthropic. Banking77 is a real-life dataset containing customer service intents and their classification labels. Contrary to other intent detection datasets, Banking77 contains a large number (77) of labels with semantic overlaps (Table 1). These characteristics make it suitable for investigating methodological perspectives while, at the same time, solving a business use case.

First, we fine-tune MPNet [38], a pre-trained Masked Language Model (MLM) by providing it with the complete dataset (Full-Data Setting with ~10k training examples). Second, we fine-tune MPNet again, but this time, we use SetFit [40], a contrastive learning technique that can achieve comparable results when shown only up to 20 examples per class (Few-Shot), instead of the complete dataset. Third, we leverage in-context learning with a wide range of popular conversational LLMs, including GPT-3.5, GPT-4, and Cohere's Command and Anthropic's Claude. We provide the LLMs with only 1 and 3 examples per class (Few-Shot), which we assume is practical from a business perspective in data-limited domains like finance [26]. In this setting, we also demonstrate that curated

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ICAIF'23, November 27-29, 2023, New York, NY

© XYZ1 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXXXX.XXXXXXX>

Financial Intent	Label
My card was declined.	Declined Card Payment
It declined my transfer.	Declined Transfer
How do you calculate your exchange rates?	Exchange Rate
My card was eaten by the cash machine.	Card Swallowed
I lost my card in the ATM.	Card Swallowed
I got married, I need to change my name.	Edit Personal Details
...	...
My card is needed soon.	Card Delivery Estimate
What is the youngest age for an account?	Age Limit

**Table 1: Example financial intents and their labels from the Banking77 dataset. In total, there are 77 different labels in the dataset with highly overlapping semantic similarities.**

samples picked by a hired human expert outperform randomly selected ones with up to 10 points difference. Overall, we show that in-context learning with LLMs can outperform fine-tuned masked language models (MLMs) in financial few-shot text classification tasks, even when presented with fewer examples. Since this solution does not require the setup of GPUs, extensive technical expertise, or a large volume of available data, it brings significant value to the financial industry, where resource constraints often exist.

Apart from studying performance, we also present a cost analysis on employing LLMs. Motivated by this, we extend our study with a cost-effective LLM inference method based on Retrieval-Augmented Generation (RAG) that reduces costs significantly, enabling small organizations to leverage LLMs without incurring substantial expenses. We demonstrate that with this approach, we retrieve only a tiny but crucial fraction (2.2%) of examples compared to the classic few-shot settings, and we are able to surpass the performance of the most competitive method (GPT-4) by 1.5 points while costing 700\$ less in the context of the test set (~3k examples). Furthermore, we simulate a low-resource data scenario, where we augment the training dataset by leveraging GPT-4, demonstrating improved performance. We also analyze the threshold where data augmentation loses effectiveness, aiding AI practitioners in their decision-making.

To the best of our knowledge, this is the first study that presents together a comprehensive evaluation of such methods in a resource-limited industrial context, where data and budget availability are key factors.

## 2 RELATED WORK

### 2.1 Studies on Banking77

Casanueva et al. [5] introduced Banking77 and achieved baseline accuracy scores of 93.6% and 85.1% for Full-Data and 10-shot settings by fine-tuning BERT [12] and using Universal Sentence Encoders [6]. Ying and Thomas [45] later improved these results by addressing label errors in the dataset, utilizing confident learning and cosine similarity approaches for mislabeled utterance detection [29]. Their trimmed dataset yielded a significant performance boost, with a 92.4% accuracy and 92.0% F1 score. Li et al. [21] demonstrated that pre-training intent representations can improve performance in financial intent classification. They achieved an 82.76% accuracy and 87.3% Macro-F1 score on the Banking77 Full-Data benchmark using prefix-tuning and fine-tuning of the last LLM layer. Lastly, Mehri and Eric [28] proposed two dialogue system text classification

Banking77 Statistics	Train	Test
Number of examples	10,003	3,080
Average length (in characters)	59.5 ± 40.9	54.2 ± 34.7
Average length (in words)	11.9 ± 7.9	10.9 ± 6.7
Number of intents (classes)	77	77

**Table 2: Banking77 dataset statistics. The average lengths are shown along with their corresponding standard deviations.**

approaches: "observers" and example-driven training. Observers offered an alternative to the [CLS] token for semantic representation and their example-driven training method leveraged sentence similarity for classification. They achieved an accuracy of 85.9% in the 10-shot setting and 93.8% in the Full-Data setting.

### 2.2 Few-Shot Text Classification

Learning from only a few training instances is crucial, especially in real-world use cases where there is no prior dataset and typically there are limited or no resources to create one. In such cases with very limited data, fine-tuning often performs poorly [13] and actually becoming more challenging as language models grow in size. An alternative approach is in-context learning [4], which involves prompting a generative large language model (LLM) with a context and asking it to complete NLP tasks without fine-tuning. The context usually includes a brief task description, some examples (the context), and the instance to be classified. The idea behind in-context learning is that the language model has already learned several tasks during pre-training, and the prompt attempts to identify the appropriate one [36]. However, selecting the right prompt is not easy as language models cannot understand the meaning of the prompt [42]. To address this issue, LLMs have been fine-tuned to follow human instructions [30, 31]. Despite this improvement, in-context learning is still correlated by term frequencies encountered during pre-training [35]. At the same time, instruct-tuned LLMs (like GPT-3.5 and GPT-4 by OpenAI), carry the biases of the human annotators who provided the training instructions. To overcome these challenges, prompt-tuning has emerged as a promising research direction [17, 19].

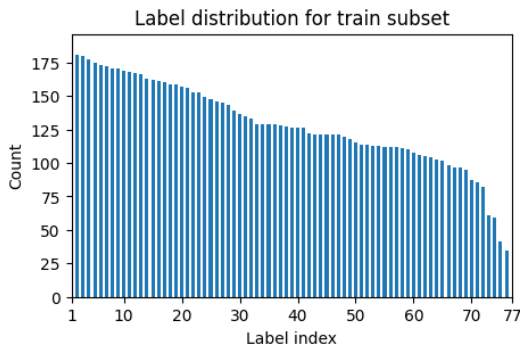
## 3 TASK AND DATASET

Intent detection is a particular case of text classification, and it is a vital component of task-oriented conversational systems in various domains, including finance. It reflects the complexity of actual commercial systems, which can be attributed to the partially overlapping intent categories, the need for fine-grained decisions, and the lack of comprehensive datasets in finance [5, 26, 27, 47].

However, publicly available intent detection datasets are limited, and the existing ones fail to represent the complexity of real-world industrial systems [3, 11]. In response to the need for industry-ready datasets [18, 24], PolyAI released Banking77 [5], which focuses on a single domain and consists of 77 fine-grained intents related to banking. By concentrating on a specific domain and offering a diverse set of intents, the dataset emulates a more realistic and challenging intent detection task than most generic benchmarks. Also, it

is worth noting that some intent categories partially overlap, requiring fine-grained decisions that cannot rely solely on the semantics of individual words, further highlighting the task’s difficulty.

The dataset comprises 13,083 annotated customer service queries labeled with 77 intents and is split into two subsets: train (10,003 examples) and test (3,080 samples) (Table 2). The label distribution is heavily imbalanced in the training subset (Figure 1), demonstrating the challenge of developing classifiers in the Full-Data setting.<sup>1</sup>



**Figure 1: Class distribution of the 77 intents in the training set. Intent indices are shown instead of tag names for brevity.**

## 4 METHODOLOGY

### 4.1 Fine-tuning MLMs

MPNet [38] is a transformer-based model [12, 41] designed with a unique pre-training objective, employing permuted language modeling to capture token dependencies and utilizing auxiliary position information as input. Pre-trained on a substantial 160GB text corpora, MPNet demonstrates superior performance compared to BERT [12], XLNet [44], and RoBERTa [25] across various downstream tasks. We utilize `all-mpnet-base-v2`<sup>2</sup>, a top-performing variation of MPNet in the sentence transformers leaderboard, making it a prominent choice for our task.<sup>3</sup>

### 4.2 Few-Shot Contrastive Learning with SetFit

SetFit [40] is a recent methodology developed by HuggingFace which fine-tunes a Sentence Transformer model on a minimal number of labeled text pairs for each class.<sup>4</sup> SetFit utilizes contrastive learning [9] in a Siamese manner, where positive and negative pairs are constructed by in- and out-class selection. As a result, transformers using SetFit produce highly descriptive text representations, where a classification head is later trained on. Despite using limited training data (such as eight training examples per class), Tunstall et al. showed that SetFit’s performance is comparable to models trained on complete datasets with standard fine-tuning [40].

### 4.3 In-Context Learning

For in-context learning, we use closed-source LLMs like **GPT-3.5** [30] and **GPT-4** [31], which are based on the Generative Pre-trained

<sup>1</sup>The test subset comprises 40 instances for every label.

<sup>2</sup><https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

<sup>3</sup>[https://www.sbert.net/docs/pretrained\\_models.html](https://www.sbert.net/docs/pretrained_models.html)

<sup>4</sup><https://github.com/huggingface/setfit>

Transformer (GPT) [33, 34]. Such recent models, after their pre-training, are additionally trained to follow instructions with Reinforcement Learning from Human Preferences (RLHF) [10, 32]. GPT-3.5 is a 175B-parameter model and, at the time of the writing, has 2 variants able to consume a context of 4,096 and 16,384 tokens. GPT-4 is a multi-modal model, with 2 variants available, able to consume 8,192 and 32,768 tokens.<sup>5</sup> We also used **Cohere’s Command** model with a context window of 4,096 tokens and **Anthropic Claude** models with a context window of 100,000 tokens.<sup>6</sup>

## 4.4 Human Expert Annotation for Robustness

Casanueva et al. [5] discovered class overlaps in creating Banking77. To address this issue, previous studies like Ying and Thomas [45] used additional annotation to curate subsets and enhance performance on real-life noisy datasets like Banking77. However, they did not share their curated subset for reproducibility.

To tackle these challenges, we curated a subset of Banking77 by hiring a subject matter expert. We provided the human expert with 10, randomly-picked examples per class, and they selected the top 3 based on their alignment with the corresponding intents. This meticulous approach reduced class overlaps and ensured the high relevance of each example to its intended label. As we show later, this is fundamental in the few-shot scenario since these expert-selected training instances outperform randomly selected instances per class. To support robust financial research, we provide this curated subset as a free resource for the financial AI community.<sup>7</sup>

## 5 EXPERIMENTS & RESULTS

### 5.1 Experimental Setup

For the **fine-tuning** methods, we use TensorFlow and HuggingFace. For all **in-context learning** methods, we instruct the model to return only the financial intent label of the test example that is presented in each query.<sup>8</sup> The prompt we use can be broken down into 3 parts and is the same for all LLMs, i.e., GPT-3.5 and GPT-4 (OpenAI),<sup>9</sup> Command (Cohere), and Claude (Anthropic). The first part contains the description of the task and the available classes. The second provides a few examples, and the third presents the text to be classified:

<sup>5</sup>OpenAI has not disclosed the architecture of GPT-4.

<sup>6</sup>See <https://cohere.com/> and <https://www.anthropic.com/>

<sup>7</sup>The curated dataset is available at [https://anonymous.4open.science/api/repo/data-prompt-4805/file/best3\\_train.csv?download=true](https://anonymous.4open.science/api/repo/data-prompt-4805/file/best3_train.csv?download=true)

<sup>8</sup>Before proceeding with closed-source LLMs, we prompted and questioned them to verify that the Banking77 dataset was not included in their pre-training data.

<sup>9</sup>We use the available gpt-3.5-turbo variants of 4K and 16K contexts for the 1-shot and 3-shot settings accordingly, and the gpt-4 8K context model, which is less expensive than the 32K one. In all cases we use the 0613 releases.

You are an expert assistant in the field of customer service. Your task is to help workers in the customer service department of a company. Your task is to classify the customer's question in order to help the customer service worker to answer the question. In order to help the worker, you MUST respond with the number and the name of one of the following classes you know. If you cannot answer the question, respond: "-1 Unknown". In case you reply with something else, you will be penalized.

The classes are:

```
0 activate_my_card
1 age_limit
.. ..
75 wrong_amount_of_cash_received
76 wrong_exchange_rate_for_cash_withdrawal
```

Here are some examples of questions and their classes:

```
How do I top-up while traveling? automatic_top_up
How do I set up auto top-up? automatic_top_up
... ..
It declined my transfer. declined_transfer
```

How do I locate my card?

## 5.2 Hyperparameter tuning in MLMs

We tuned the MLM (MPNet-v2) using Optuna’s [1] implementation of the Tree-structured Parzen Estimator (TPE) algorithm [2]. We specified 10 trials, and we defined a search space of (1e-5, 5e-5) for the body’s learning rate and (1e-2, 5e-5) for the head’s learning rate with logarithmic intervals. During tuning, we maximized the validation Micro-F1. We deployed the MLM pipelines in an NVIDIA A100 SXM GPU with 40GB memory.

## 5.3 Prompt Engineering in LLMs

We experiment with two different prompt settings using GPT-4 in a 3-shot setting on a held-out validation subset, which was created by using 5% of the training subset. In the first setting, we include the few-shot examples as part of the chat history, i.e., the query is presented as a user message and the class is presented as an assistant message that follows. In the second setting, we include the few-shot examples in the so-called system, which is intended to give an initial context to the assistant. The second setting yielded the best results (Table 3), and we proceed to use it for the rest of our experiments.

Few-shot examples given as	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Previous chat history	75.5	74.4
System context	<b>77.7</b>	<b>77.0</b>

**Table 3: Validation Micro-F1 and Macro-F1 scores for our two prompt settings with GPT-4 in the 3-Shot scenario.**

Methods	Setting	$\mu$ -F <sub>1</sub>	m-F <sub>1</sub>
Mehri and Eric [28]	Full-Data	93.8	NA
Mehri and Eric [28]	10-shot	85.9	NA
Ying and Thomas [45]	Full-Data	NA	92.0
MPNet-v2	Full-Data	<b>94.1</b>	<b>94.1</b>
MPNet-v2 (SetFit)	1-shot	57.4	55.9
Cohere Command	1-shot	44.9	46.5
Anthropic Claude	1-shot	73.8	72.1
GPT-3.5 (representative samples)	1-shot	<b>75.2</b>	<b>74.3</b>
GPT-3.5 (random samples)	1-shot	74.0	72.3
MPNet-v2 (SetFit)	3-shot	76.7	75.9
GPT-3.5 (representative samples)	3-shot	65.5	65.3
GPT-4 (representative samples)	3-shot	<b>83.1</b>	<b>82.7</b>
GPT-4 (random samples)	3-shot	74.2	73.7
MPNet-v2 (SetFit)	5-shot	83.5	83.3
MPNet-v2 (SetFit)	10-shot	88.1	88.1
MPNet-v2 (SetFit)	15-shot	90.6	90.5
MPNet-v2 (SetFit)	20-shot	91.2	91.3

**Table 4: Classification results for all models on the test data, with N-Shot indicating the number of samples used during training. The MPNet model is fine-tuned without the SetFit method on the Full-Data setting.**

## 5.4 Results

To comprehensively understand how accurately the models can perform, we report micro-F1 ( $\mu$ -F<sub>1</sub>) and macro-F1 (m-F<sub>1</sub>). Table 4 shows MPNet-v2 achieving competitive results across all few-shot settings using SetFit. When trained on only 3 samples, it achieves scores of 76.7  $\mu$ -F<sub>1</sub> and 75.9 m-F<sub>1</sub>. As we increase the number of samples, the performance improves, reaching a 91.2 micro-F1 and 91.3 macro-F1 score with 20 samples. This is only 3 percentage points (pp) lower than fine-tuning the model with all the data, demonstrating the effectiveness of SetFit, especially in domains where acquiring data points is difficult. Lastly, our MPNet-v2 solution outperforms the previous results from [28], both in the 10-shot and Full-Data settings (by 2.2 pp and 0.3pp).

Despite being presented with only 1 sample per class (either random or representative), GPT-3.5 achieves competitive results (74.0 and 75.2  $\mu$ -F<sub>1</sub>). It outperforms MPNet-v2 by a large margin (over 17 pp) in this 1-shot setting, showing the potential for effective few-shot text classification in the financial domain where data are not always readily available [26]. As expected, using our human-curated representative samples leads to better in-context learning results. We also employed two alternative closed-source LLMs by Cohere and Anthropic on the representative samples. Cohere’s Command performs poorly with a low 44.9  $\mu$ -F<sub>1</sub> score while Anthropic’s Claude with a 73.8  $\mu$ -F<sub>1</sub> performs comparable to GPT-3.5’s 75.2  $\mu$ -F<sub>1</sub>.

GPT-4 also shows potential for few-shot text classification, outperforming both previous models on the 3-shot setting by more than 6 pp (MPNet-v2) and 17 pp (GPT-3.5). More notably, the less powerful engine of GPT-3.5, performs poorly on its 3-shot variant (65.5  $\mu$ -F<sub>1</sub>) compared to its promising 1-shot variant (75.2  $\mu$ -F<sub>1</sub>). This probably verifies recent reports on GPT-3.5 getting lost in the middle of long contexts [23]. Similarly to the 1-shot experiments

with GPT-3.5, the performance of GPT-4 drops substantially (approximately 9 pp) when trained on random samples as opposed to when trained on the human-curated representative ones. Thus, at least in this complex financial text classification task, it is better to present more examples to a more powerful engine like GPT-4 instead of GPT-3.5.

## 6 COST-EFFECTIVE LLM INFERENCE USING RETRIEVAL-AUGMENTED GENERATION

### 6.1 Cost Analysis

Apart from studying the models' performance, we investigate the significant Operating Costs (OpEx) associated with popular LLMs like GPT-4. We provide a budget analysis for our experiments in Table 5 (cost refers to ~3k instances). This can be seen as a guideline for researchers and practitioners to evaluate the trade-offs between performance and budget when selecting a closed-source LLM of-fered via a managed API, as well as a data point when deciding between a "build vs. buy" approach, which requires developing datasets and setting up and hosting their own custom model.

Model	Setting	Micro-F1↑	Cost↓
GPT-3.5	1-shot	75.2	31\$
Anthropic Claude	1-shot	73.8	15\$
Cohere command	1-shot	44.9	22\$
GPT-3.5	3-shot	65.5	62\$
GPT-4	3-shot	83.1	740\$
GPT-4	5 similar (RAG)	84.5	205\$
Anthropic Claude	5 similar (RAG)	84.8	33\$
GPT-4	10 similar (RAG)	81.2	230\$
Anthropic Claude	10 similar (RAG)	85.2	37\$
GPT-4	20 similar (RAG)	87.7	270\$
Anthropic Claude	20 similar (RAG)	85.5	42\$

**Table 5: Cost analysis of various closed-source LLMs. The first two groups represent the 1- and 3-shot results, as shown in Section 5.4 (Results). The rest groups come from Section 6.2, where we utilize Retrieval-Augmented Generation (RAG) for cost-effective LLM inference. We perform 3,080 queries to each LLM, i.e., 1 query per sample in the test set.**

Focusing on the two upper groups of the table, we see how popular LLMs perform in the financial intent detection task of Banking77 in a standard few-shot approach. We observe that in the 1-shot setting, Anthropic Claude has a performance that comes on par with the GPT-3.5 model and comes at half of GPT-3.5's cost. In the 3-shot setting, GPT-4 scores nearly 20 points more than GPT-3.5, but comes with a 10x cost, that amounts to 740\$.<sup>10</sup>

### 6.2 Retrieval-Augmented Generation (RAG)

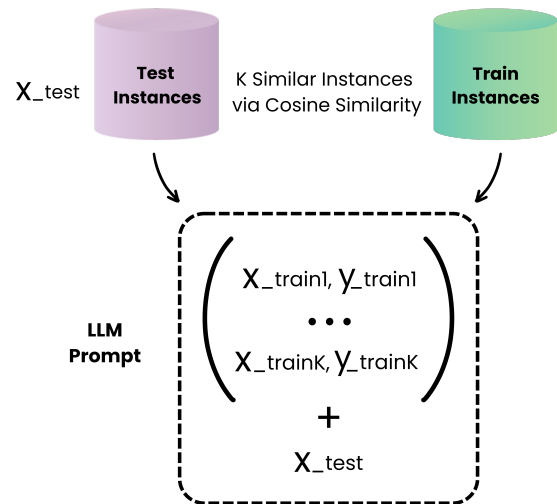
Motivated by the high costs associated with closed-source LLMs, we present a methodology to query LLMs efficiently by creating a

<sup>10</sup>We were motivated to experiment with Anthropic's Claude in a 3-shot setting since it showed great potential in the 1-shot setting, but our budget for experimentation was limited. Thus, we proceeded with the most promising models as our "best bets".

Retriever component before plugging it into our Generative Text Classifier (Reader) pipeline.

In the standard few-shot setting, we provide examples for all intents; e.g., in the 3-shot setting, we use a prompt with 231 samples (3 samples x 77 classes) as context in each inference call to the LLMs. However, our intuition is that a subset of them can be sufficient, allowing room for a cost-effective approach by narrowing down the number of representative examples used in each inference call.

Thus, we consider including only the most similar examples to be utilized during each inference call. This is based on the active learning algorithms proposed by Lewis et al. and Liu et al., who augment in-context examples for text generation language models using kNN. The rationale is: for each inference sentence to be classified, instead of providing all the 231 examples for the classes inside the prompt, we provide only the 5 (2.2%), 10 (4.3%), and 20 (8.7%) most similar examples, as shown in Figure 2. We use the all-mpnet-base-v2 vector embeddings and the cosine similarity metric for the distance calculation.



**Figure 2: Dynamic LLM prompt construction through Retrieval-Augmented Generation (RAG), using cosine similarity for in-context data selection. We use  $K=5, 10, 20$ .**

The results of this approach, as seen in the three lower groups of Table 5, show improved results both in performance and budget. For instance, when we retrieve only five similar instances (2.2% of the 231 total examples) for each inference call, Anthropic's Claude has an 84.8  $\mu$ -F<sub>1</sub> vs. GPT-4's 84.5. At the same time, it costs 172\$ less (1/6 of GPT-4's cost). Most importantly, this score is also higher than 1.5 percentage points than GPT-4 in the classic 3-shot setting, which costs 740\$ and is 22 times multiple of its cost.

By increasing the retrieved similar samples to 10 and 20, we see a slight increase to a maximum of 87.7%  $\mu$ -F<sub>1</sub> for GPT-4 (270\$) and 85.5%  $\mu$ -F<sub>1</sub> of Claude (42\$). Anthropic's Claude is consistent in improving over the number of retrieved examples, while GPT-4 shows a slight drop in performance when presented with 10 similar examples vs. 5 similar examples. One possible explanation for this behavior is that it may be due to the specific LLM checkpoints' volatility in results, as reported very recently in various tasks [8].

## 7 LLMs FOR DATA GENERATION IN LOW-RESOURCE SETTINGS

Data-centric AI is increasingly gaining attentions as evidenced by relevant scientific venues.<sup>11</sup> Such approaches become invaluable in areas where collecting vast quantities of data is either infeasible, prohibited by privacy constraints, or challenging such as in finance.<sup>12</sup> They focus on maximizing the utility of limited datasets rather than only focusing on scaling up model complexity (model-centric AI), which is often impractical to maintain or deploy [15, 16].

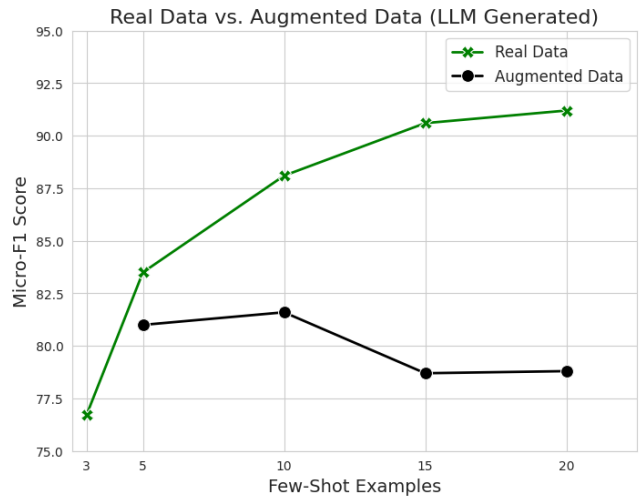
In line with this low-resource paradigm, we simulate a scenario where our dataset is limited. We assume that providing 3 examples per class is feasible in practice, we start with the 231 original hand-picked examples from the expert (3 per each of the 77 classes), and we leverage GPT-4 to generate additional examples per class. The intention here is to explore how much we can rely to data augmentation to increase the performance, and identify the threshold beyond which the quality of generated examples starts to degrade, and thus, we cannot continue with adding more (artificial) examples.

While past existing studies show that just writing simple prompts boost performance on various classifications tasks, they come short due to two reasons: (a) they consider vanilla tasks with up to 6 classes (where in our scenario, we consider 77) [46] and (b) they use older text generation models rather than instruct-based LLMs.

Since Sahu et al. found that data augmentation for tasks with large label volume is likely to not benefit at all (at least when using older text generation models), we adopt a particular prompt template for our instruct-based GPT-4 model. First, we analyze the data using Azimuth [14], an open-source toolkit. We split the 77 labels into ten groups, where each group consists of labels that contain intents with highly semantic overlap, as shown in Azimuth (e.g., Top Up Reverted and Top Up Failed belong to the same group). Then, we prompted the model with the 3 examples of each class inside that group, and we specifically requested it to generate 20 artificial examples by paying attention to these confounding classes.<sup>13</sup>

Subsequently, we employed the SetFit approach across four different settings: 5-shot, 10-shot, 15-shot, and 20-shot. However, to incorporate the generated data, we amended the traditional methodology of presenting the N original examples per class. Instead of the 5-shot experiment, we used 3 examples from the original data and 2 from the augmented. Similarly, the original-to-augmented ratios for 10-shot, 15-shot, and 20-shot tasks were 3:7, 3:12, and 3:17, respectively. This allowed us to assess how well our model performs with limited data and how effectively it can integrate and learn from artificially generated examples.

Figure 3 shows the  $\mu$ -F<sub>1</sub> score for different few-shot settings. Starting from the 3-shot setting with 76.7%  $\mu$ -F<sub>1</sub>, one can generate artificial data to get an essential boost to 81%  $\mu$ -F<sub>1</sub> as seen in the augmented 5-shot setting (3 original examples + 2 generated). The maximum performance increase is observed at the 10-shot augmentation scenario with 81.6%. After this threshold, the quality of the data generated decreases (see 15- and 20-shot settings), injecting more noise into the model than the actual value. As expected, the



**Figure 3: The Micro-F1 Score for various few-shot settings. In the Augmented Data (black) line, 3 of the examples each time (out of the 5, 10, 15, 20) belong to the representative samples that the human expert picked from the real data.**

real data (even if they are challenging to obtain in such domains) are far superior and more meaningful than the generated data (91.2  $\mu$ -F<sub>1</sub> vs. 78.8  $\mu$ -F<sub>1</sub> in the 20-shot setting).

## 8 ERROR ANALYSIS

Lastly, to understand our models' limitations, we also performed an exploratory error analysis in some of our top-performing models. Specifically, we manually inspected the errors of GPT-3.5 (1-shot), GPT-4 (3-shot), MPNet-v2 (10-shot), and MPNet-v2 (Full-Data).

### 8.1 Errors in LLMs

The most misclassified labels by GPT-4 (3-shot) and GPT-3.5 (1-shot) are shown in Table 6 along with their percentages. After inspecting the samples and their predicted labels, we observe that the label Get Physical Card was frequently misclassified as Change Pin. This is potentially due to the presence of the word "PIN" in all test instances, causing the vector embeddings to locate them close to the decision boundary of the Change Pin class. As for the Transfer Not Received By Recipient class, phrases indicative of non-receipt were observed in only 20% of the test samples, which could have complicated the classification task. The rest of the samples exhibited concerns related to transaction timing and recipient visibility, pushing them towards the Transfer Timing class.

Gold Label	Misclassifications	
	GPT-4	GPT-3.5
Get Physical Card	87.5%	87.5%
Transfer Not Received	62.5%	50.0%
Beneficiary Not Allowed	42.5%	60.0%

**Table 6: Top misclassified labels, along with their misclassification percentages (out of 40 test instances), for the GPT-4 (3-shot) and GPT-3.5 models (1-shot)**

<sup>11</sup><https://datacentricai.org/neurips21/>

<sup>12</sup><https://sites.google.com/view/icaif-synthetic/home>

<sup>13</sup>Our prompt template for data generation can be found in <https://anonymous.4open.science/api/repo/data-prompt-4805/file/prompt.md?download=true>.



The third label presented in the table, Beneficiary Not Allowed, was incorrectly classified mostly as Declined Transfer by GPT-4 and GPT-3.5, respectively. The incorrect labeling could be attributed to sentences that might seem ambiguous even for a human classifier. For instance, sentences such as “*I tried to make a transfer, but it was declined.*” and “*The system does not allow me to transfer funds to this beneficiary.*” were misinterpreted as pertaining to Declined Transfer class, due to the mention of terms like “*declined*” and “*does not allow*”. The models, thus, failed to discern the actual issue of beneficiary disallowance.

## 8.2 Errors in MLMs

Gold Label	Misclassifications	
	Full-Data	10-shot
Top Up Failed	22.5%	17.5%
Transfer Timing	20.0%	45.0%

**Table 7: Misclassification percentages (out of 40 test instances) for the MPNet-v2 model when trained on Full-Data vs. when trained on 10 samples per class.**

In Table 7, the MPNet-v2 model displayed significant confusion between in two labels when trained on Full-Data and 10 samples per class. Specifically, after inspection, we see that for the gold label Top Up Failed, the sentence “*please tell me why my top-up failed*” was misclassified as Top Up Reverted, likely due to contextual overlap between failed and reverted top-ups. Another source of confusion involved the Transfer Timing gold label. Sentences like “*How many days does it take until funds are in my account?*” were misclassified as Pending Transfer or Balance Not Updated After Cheque or Cash Deposit. The temporal aspect of the query seems to direct the models towards other classes that also involve time, thus highlighting challenges in differentiating specific concerns related to money transfers.

## 8.3 About Overlapping Labels

We anticipate that this analysis will serve as a source of inspiration for future work in financial intent detection. One possible approach to mitigate these errors involves the adoption of hierarchical classifiers [7]. These classifiers could initially identify classes within broad categories, such as transfers or top-ups, and subsequently classify them into more specific classes, such as pending transfer vs. transfer timing and top-up failed vs. top-up reverted. We believe that these methodologies could either be integrated with “native” TensorFlow/PyTorch pipelines or simulated using Chain-of-Thought prompting techniques [43] so the LLMs can distinguish fine-grained financial labels by thinking step-by-step or class-by-class.

## 9 CONCLUSION AND FUTURE WORK

We conducted a comprehensive few-shot text classification study using LLMs, MLMs and discussed their trade-offs between cost and performance. We focused on Banking77, a financial intent detection dataset with real-life challenges, such as its large number of intents and overlapping labels.

Our results clearly demonstrate the effectiveness and efficiency of in-context learning using conversational LLMs. This approach serves as a practical and rapid solution for achieving accurate results in few-shot scenarios and domains with restricted resources, like finance. In detail, we demonstrated that LLMs, like GPT-3.5, GPT-4 and Anthropic Claude, can perform better than MLMs in some limited-data scenarios (1- and 3-shot). On the other side, by fine-tuning custom MLMs like MPNet-v2 with SetFit, we surpassed the previous work of [28] in the 10-shot setting by 2.2 pp.

While LLM services minimize the technical expertise needed or omit GPU training times, they can be considered costly for small organizations, given that they are accessed behind paywalls (we spent 740\$ for 3-shot classification of ~3k samples with GPT-4). After presenting detailed pricing costs to help the community make better decisions, we also demonstrated a cost-effective inference method for LLMs. We utilized a semantic similarity retriever to return only a small but substantial fraction of training examples to our prompt, showing that this performs better than classic few-shot in-context learning. Most importantly, using Claude with this approach, we can save multiple (22) times the cost associated with LLM services like GPT-4 (700\$ less) while getting a higher classification score.

To showcase how financial companies can utilize LLMs in scenarios with limited data, we also used GPT-4 to generate artificial data for data augmentation. We conclude that these generated data points are helpful up to a specific threshold (7 generated examples), after which our performance starts to drop, possibly due to the LLM starting to generate noise.

Lastly, all of our top-performing Few-Shot experiments using LLMs and MLMs were trained on representative data samples out of a human expert-curated Banking77 subset. We provide this curated dataset freely available in order to promote the development of robust financial AI systems.

In future work, we plan to experiment with open-sourced LLM alternatives, which may be suitable substitutes for closed-source models, like LLaMA2 [39] and incorporating Chain-of-Thought [43] techniques to mitigate the errors about the overlapping class labels, as presented extensively in the error analysis section.

## REFERENCES

- [1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 2623–2631. <https://doi.org/10.1145/3292500.3330701>
- [2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (Eds.), Vol. 24. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf)
- [3] Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the 18th Annual SIGDial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Saarbrücken, Germany, 174–185. <https://doi.org/10.18653/v1/W17-5522>
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing*

- Systems (Vancouver, BC, Canada) (*NIPS'20*). Curran Associates Inc., Red Hook, NY, USA, Article 159, 25 pages.
- [5] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics, Online, 38–45. <https://doi.org/10.18653/v1/2020.nlp4convai-1.5>
  - [6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, 169–174. <https://doi.org/10.18653/v1/D18-2029>
  - [7] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An Empirical Study on Large-Scale Multi-Label Text Classification Including Few and Zero-Shot Labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7503–7515. <https://doi.org/10.18653/v1/2020.emnlp-main.607>
  - [8] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's behavior changing over time? [arXiv:2307.09009](https://arxiv.org/abs/2307.09009) [cs.CL]
  - [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 149, 11 pages.
  - [10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf)
  - [11] Alice Coucke, Alaa Saade, Adrien Bal, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. [ArXiv abs/1805.10190](https://arxiv.org/abs/1805.10190) (2018).
  - [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
  - [13] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. [arXiv:2002.06305](https://arxiv.org/abs/2002.06305) [cs.CL]
  - [14] Gabrielle Gauthier-melancon, Orlando Marquez Ayala, Lindsay Brin, Chris Tyler, Frederic Branchaud-charron, Joseph Marinier, Karine Grande, and Di Le. 2022. Azimuth: Systematic Error Analysis for Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Abu Dhabi, UAE, 298–310. <https://aclanthology.org/2022.emnlp-demos.30>
  - [15] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *Int. J. Comput. Vision* 129, 6 (jun 2021), 1789–1819. <https://doi.org/10.1007/s11263-021-01453-z>
  - [16] Institute of Electrical and Electronics Engineers (IEEE) 2019. *2019 2nd Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2 2019)*. Institute of Electrical and Electronics Engineers (IEEE), Curran Associates, Inc., Washington, DC, USA. <https://doi.org/10.1109/EMC249363.2019>
  - [17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual Prompt Tuning. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 709–727.
  - [18] Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. 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)*. Association for Computational Linguistics, Hong Kong, China, 1311–1316. <https://doi.org/10.18653/v1/D19-1131>
  - [19] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3045–3059. <https://doi.org/10.18653/v1/2021.emnlp-main.243>
  - [20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.
  - [21] Xianzhi Li, Will Aitken, Xiaodan Zhu, and Stephen W. Thomas. 2022. Learning Better Intent Representations for Financial Open Intent Classification. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 68–77. <https://aclanthology.org/2022.finnlp-1.8>
  - [22] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics, Dublin, Ireland and Online, 100–114. <https://doi.org/10.18653/v1/2022.deelio-1.10>
  - [23] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. [arXiv:2307.03172](https://arxiv.org/abs/2307.03172) [cs.CL]
  - [24] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*. Springer Singapore, Singapore, 165–183. [https://doi.org/10.1007/978-981-15-9323-9\\_15](https://doi.org/10.1007/978-981-15-9323-9_15)
  - [25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. [ArXiv abs/1907.11692](https://arxiv.org/abs/1907.11692) (2019).
  - [26] Lefteris Loukas, Manos Fergadiotis, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. EDGAR-CORPUS: Billions of Tokens Make The World Go Round. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 13–18. <https://doi.org/10.18653/v1/2021.econlp-1.2>
  - [27] Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. 2022. FiNER: Financial Numeric Entity Recognition for XBRL Tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 4419–4431. <https://doi.org/10.18653/v1/2022.acl-long.303>
  - [28] Shikib Mehri and Mihail Eric. 2021. Example-Driven Intent Prediction with Observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 2979–2992. <https://doi.org/10.18653/v1/2021.naacl-main.237>
  - [29] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2021. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research (JAIR)* 70 (2021), 1373–1411.
  - [30] OpenAI. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf)
  - [31] OpenAI. 2023. GPT-4 Technical Report. [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL]
  - [32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.
  - [33] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding with Unsupervised Learning. <https://openai.com/research/language-unsupervised> Accessed: 06 May 2023.
  - [34] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
  - [35] Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of Pretraining Term Frequencies on Few-Shot Numerical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 840–854. <https://aclanthology.org/2022.findings-emnlp.59>
  - [36] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (Yokohama, Japan) (CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 314, 7 pages. <https://doi.org/10.1145/3411763.3451760>
  - [37] Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data Augmentation for Intent Classification with Off-the-shelf Large Language Models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*. Association for Computational Linguistics, Dublin, Ireland, 47–57. <https://doi.org/10.18653/v1/2022.nlp4convai-1.5>
  - [38] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-Training for Language Understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*



- (Vancouver, BC, Canada) (*NIPS'20*). Curran Associates Inc., Red Hook, NY, USA, Article 1414, 11 pages.
- [39] Hugo Touvron and Meta GenAI. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]
- [40] Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. *ArXiv abs/2209.11055* (2022).
- [41] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All you Need. *Advances in neural information processing systems*, 5998–6008.
- [42] Albert Webson and Ellie Pavlick. 2022. Do Prompt-Based Models Really Understand the Meaning of Their Prompts?. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 2300–2344. <https://doi.org/10.18653/v1/2022.naacl-main.167>
- [43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- [44] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Curran Associates Inc., Red Hook, NY, USA.
- [45] Cecilia Ying and Stephen Thomas. 2022. Label Errors in BANKING77. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*. Association for Computational Linguistics, Dublin, Ireland, 139–143. <https://doi.org/10.18653/v1/2022.insights-1.19>
- [46] Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2225–2239. <https://doi.org/10.18653/v1/2021.findings-emnlp.192>
- [47] Elias Zavitsanos, Dimitris Mavroeidis, Konstantinos Bougiatiotis, Eirini Spyropoulou, Lefteris Loukas, and Georgios Paliouras. 2022. Financial Misstatement Detection: A Realistic Evaluation. In *Proceedings of the Second ACM International Conference on AI in Finance (Virtual Event) (ICAFIF '21)*. Association for Computing Machinery, New York, NY, USA, Article 34, 9 pages. <https://doi.org/10.1145/3490354.3494453>