Why So Separate: Analyzing In-Context Learning from a Vector Space Perspective

Tobias Kalmbach, Sandipan Sikdar
L3S Research Center
Leibniz University Hannover
Hannover, Germany
{tobias.kalmbach, sandipan.sikdar}@l3s.de

Abstract

In-context learning (ICL) is a popular prompting strategy for large language models. ICL allows models to learn tasks using demonstrative examples alone, without any weight updates or training. Nevertheless, it is still largely unclear why ICL works. In this paper, we investigate ICL from a new viewpoint, namely a vector space perspective, and extract insights for ICL from this analysis. In our experiments, we extract the hidden representations, i.e., embeddings, created by a large language model when passing an ICL prompt through it. We find that these embeddings generated by large language models are separable in the vector space when applying ICL. The degree of separability is dependent on the difficulty of the task, the size of the model and other factors, like the labels of demonstrative examples. We also find that, especially for large models, the separability is indicative of the classification performance. As an application, we utilize our findings to explain peculiarities of ICL and to select demonstrative examples for ICL. Experiments across multiple datasets show that this way of selecting examples consistently outperforms the commonly used random selection method.

Keywords: In-context learning, Large language models, Interpretability

1. Introduction

Few-shot and zero-shot learning have demonstrated remarkable capabilities in large language models (LLMs) for many applications (Long et al., 2024b,a; Dong et al., 2024). Few-shot and zero-shot learning, also called in-context learning (ICL) (Brown et al., 2020), are prompting paradigms, generally improving performance for LLMs. With ICL, pre-trained models only need the demonstrative examples to adapt to unseen tasks, eliminating the need for task-specific training data and expensive model updates. The task is learned solely using demonstrative examples in the input prompt. For instance, given a few examples of sentiment classification like "This movie was great! — Positive" and "I did not like the food. — Negative", followed by an unseen sentence, LLMs can accurately classify the sentiment despite never being explicitly trained for this task.

Several studies have shown the effectiveness of this paradigm, yet the underlying mechanisms *why* ICL works remain poorly understood (Li et al., 2024b; Tang et al., 2024). As models can adapt to new tasks, this suggests that LLMs develop some meta-learning capabilities during pretraining, enabling pattern recognition and alignment during inference based on demonstrative examples only (Wei et al., 2022b). Nonetheless, the precise mechanisms underlying this adaptation remain debated in the research community.

In this paper, we provide an alternative analysis of ICL through the lens of vector space geometry, examining how LLMs process and represent ICL prompts in their hidden states[1]. An overview of our analysis is shown in Figure 1.

LLMs are neural networks based on the Transformer architecture (Vaswani et al., 2017), comprising multiple layers of self-attention and feed-forward neural networks. For an input, the text is tokenized and embedded into a high-dimensional vector space. Typically, the early layers encode basic linguistic features like syntax and relationships, while deeper layers focus on semantic representations and task-specific features (Rogers et al., 2020). Moreover, the output of the hidden layers corresponding to the last token represents an encoding of the processed version of the complete input.

By extracting and visualizing embeddings for the last token, we find that for classification tasks, LLMs can produce embeddings that are clustered based on class distinctions. We also find that classification accuracy, in general, is higher with better clustering (see Figure 1(d)). This clustering is, however, specific to larger models ($>$10B parameters) and is not necessarily observed for smaller models. Such clustering in the embedding space suggests that as model scale increases, LLMs develop more structured internal representations of task-relevant features during ICL.

Building on these insights, we aim to explain peculiarities observed for ICL (Min et al., 2022; Shi et al., 2024) and we propose a method for

[1] Code: https://github.com/Tobi2K/Why-So-Separate

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 93–106
11–16 May 2026.

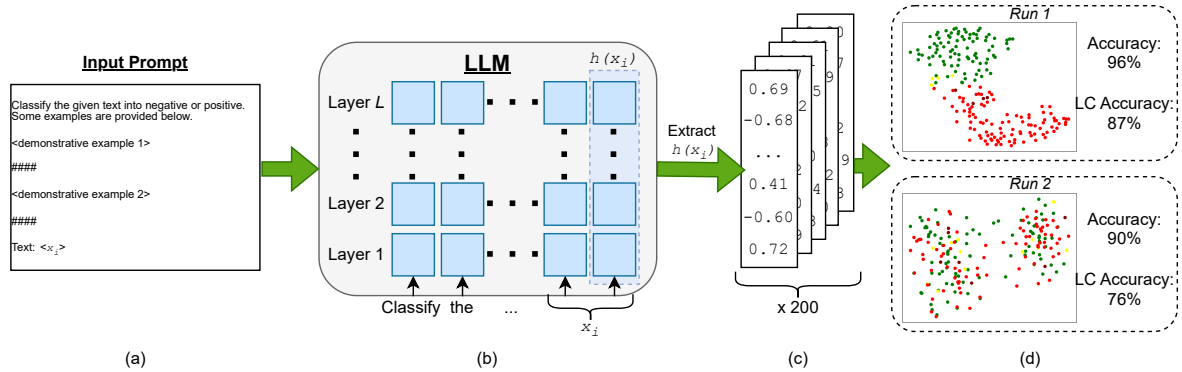


Figure 1: Overview of our method. (a) When passing an inference instance x_i with the shown prompt through the model, (b) we extract the hidden representation $h(x_i)$ corresponding to the last token of x_i. (c) We repeat this for several x_i and (d) apply t-SNE to visualize and a linear classifier (LC) to analyze the hidden representations. The two runs (Run 1 and Run 2) correspond to two different sets of demonstrative examples in the prompt, and hence also the difference in embeddings of x_is for the two runs. We find that runs with well-separated embeddings (Run 1) achieve higher classification and linear classifier accuracy than poorly separated embeddings (Run 2).

selecting demonstrative examples based on their degree of separability in the vector space. This method achieves better ICL performance than random sampling, which is usually deployed in many applications. Our work sheds light on why some demonstrative examples work better than others. Further, this vector space perspective is a simple and intuitive view that uses the model's internal representations directly instead of employing a proxy.

2. Related Work

The phrase "in-context learning" for LLMs was first introduced by Brown et al. (2020), although the technique was already used previously (Radford et al., 2019). In-context learning (ICL) has since been used for various tasks, including, but not limited to, NLP tasks (Zhu et al., 2024a), embedding generation (Li et al., 2024a), even computer vision (Zhang et al., 2023) and many others (Nafar et al., 2024; Chen et al., 2024; Li et al., 2023; Zhou et al., 2023; Mohamed et al., 2025; Hegde et al., 2025).

Why ICL works (and when or why it does not) is still fundamentally unanswered (Dherin et al., 2025). Some investigations point out that adding a context improves task performance (Min et al., 2022), while others explicitly point to the opposite (Li et al., 2024b). Other studies show that selecting *too* similar examples does not improve performance (Fu et al., 2024) and may not show the breadth of possible classes (D'Oosterlinck et al., 2024). Some works point to changes in the formatting of examples impacting performance (Tang et al., 2024), although different models work best with different formatting (Voronov et al., 2024). On the other hand, Min et al. (2022) show that replacing ground truth labels with random labels only slightly worsens performance. Nevertheless, ICL can be highly sensitive to various parameters like the ordering of examples in the prompt (Chang and Jia, 2023; Lu et al., 2022), but also model configurations like the chosen decoding strategy (Ling et al., 2024). Further, other works have shown that ICL is also unreliable (i.e., can both increase and decrease performance) for reasoning tasks (Liu et al., 2025; Alazraki et al., 2025). All these investigations indicate that *there isn't a single recipe to make ICL work*.

ICL, depending on the task, can outperform traditional instruction tuning (e.g., SFT and RLHF) in LLMs (Wang et al., 2024; Lin et al., 2024). Recent works have found that ICL can act as gradient descent optimization in the forward pass (Ahn et al., 2023; von Oswald et al., 2023a,b; Akyürek et al., 2023; Dai et al., 2023). However, Deutch et al. (2024) argue that most of the *analyses above are too simple or insufficient to confidently claim that ICL acts as gradient descent*.

Selecting Demonstrative Examples Most studies default to using randomly selected examples, but options like ordering or formatting of demonstrative examples can impact the performance (Chen et al., 2023; Lu et al., 2022; Min et al., 2022; Zhu et al., 2024b). One group of selection frameworks that does not use random sampling uses ranking metrics to select examples that influence the prediction of the LLM the most (Wu et al., 2023; Guo et al., 2024; M.S. et al., 2024). Other approaches let the model rank the demonstrative examples itself (Yao et al., 2024) or train an encoder to summarize inference instances and select similar demonstrative examples (Gupta et al., 2024). Another group of works uses external models to rank or predict the effectiveness of candidate exam-

ples (Long et al., 2024a; Qian et al., 2024; Wang et al., 2023; Xie et al., 2022). Other works (Li and Qiu, 2023; Gao et al., 2024) use filtering or compression to more efficiently select examples from a large set.

Present work While all prior investigations have their merit, our approach analyzes demonstrative examples solely from a vector space perspective, which is simpler and more visually intuitive. As Deutch et al. (2024) point out, analyses claiming that ICL is equivalent to gradient descent are insufficient. We are therefore motivated to view ICL from a different perspective. The proposed vector space perspective is a simple and, using t-SNE visualization, intuitive approach that can not only be used to analyze ICL itself but can also be extended to use cases like demonstrative example selection or ranking. Lastly, our method also allows for interpreting the idiosyncrasies observed in ICL, e.g., replacing ground truth labels with random labels only slightly worsens performance (Min et al., 2022).

Our methodology is partly inspired by ICV ("In-Context Vectors") proposed by Liu et al. (2024). However, they are primarily interested in improving ICL performance across tasks rather than explaining ICL performance.

3. Methodology

In the following, we refer to the examples added to the prompt as demonstrative examples. Unless otherwise specified, we refer to the setup as in-context learning when using one or more demonstrative examples and zero-shot when no examples are specified in the prompt. Additionally, the demonstrative examples are preceded by a short instruction describing the task. This instruction is task-dependent, e.g., "Classify the given text into negative or positive." for sentiment classification or "Decide whether sentence 1 is equivalent to sentence 2." for semantic equivalence. When adding n demonstrative examples, we also refer to this setup as n-shot learning, and we add $\frac{n}{2}$ examples of class 0 and $\frac{n}{2}$ examples of class 1. In the default case, we randomly select two demonstrative examples of each class.

Proposed Method Our methodology builds on two observations: (i) LLMs encode features linearly, even for non-linear applications (Nanda et al., 2023; Jiang et al., 2024), and (ii) later (upper) layers encode high-level concepts (Elhage et al., 2022; Gurnee et al., 2023). In fact, Hollinsworth et al. (2024) show that sentiment is a concept that is stored linearly, representing positive as one extreme and negative as the other extreme. Based on the above studies, we hypothesize that such linearity should also be observed in the hidden representations of demonstrative examples.

We consider only classification tasks in this work. In this setup, we are given input sentences x, and the task is to predict the corresponding class. In the following, we will focus on tasks with two classes, but the setup can be extended to more classes. We illustrate our approach in Figure 1. We create a prompt P with an instruction about the task, followed by a set of demonstrative examples. For a given inference instance x, we append it to P (Figure 1(a)) and we copy the embedding generated in the model corresponding to the last token (Figure 1(b)), which essentially corresponds to the last token of x. Note that this embedding of x is specific to the demonstrative examples used in the prompt and would differ as we change the demonstrative examples. This process is repeated for several inference instances (200 in our case), resulting in a distinct embedding corresponding to each instance (Figure 1(c)). In Figure 1 (d), we consider two different runs, each consisting of a different set of demonstrative examples. Depending on the selected demonstrative examples, the orientation of the embeddings changes: one case (top) leads to embeddings where the instances of the two classes are linearly *separable*, while for the other (bottom), it is not so.

Separability Our overall hypothesis is that the performance of ICL for a given task is directly associated with *separability*, i.e., the ability to linearly discriminate between the instances of the constituent classes. Formally, we define separability as *the ability to split instances of a dataset, e.g., by class, using the vector space location of the hidden representations.* The extent of separability is further dependent on the demonstrative examples, as each text instance is appended to the prompt containing the demonstrative examples. Therefore, it is imperative to select demonstrative examples carefully, as using random examples can hinder the internal separability of hidden representations.

The extent of separability can be visualized using methods such as t-SNE (van der Maaten and Hinton, 2008), which allows for visualizing higher-dimensional data in two dimensions. However, t-SNE has drawbacks. The shown distances do not necessarily correspond to closeness in high-dimensional space, the visualization is invariant to rotations and translation, and the compression to two dimensions can result in a loss of information, including separability. We use t-SNE as an easy and appealing showing of the separability. We only provide t-SNE visualization as a motivation and always supplement them with the accuracy score

of a linear classifier[2] fit on the embeddings, which is more reliable in terms of separability. Higher accuracy indicates higher separability and vice versa. We note for the reported accuracies that despite a low linear classifier accuracy, the data may be separable in higher dimensions. A low score can still, for example, happen if the linear classifier learned a distribution that diverges from the tested data.

4. Experimental Apparatus

4.1. Datasets

We use five datasets comprising four tasks. The tasks are sentiment analysis (*IMDb* (Maas et al., 2011) and *SST-2* (Socher et al., 2013)), textual entailment (*RTE* (Wang et al., 2018)), grammatical acceptability (*CoLA* (Warstadt et al., 2019)), and semantic equivalence (*MRPC* (Dolan and Brockett, 2005)). For IMDb, we select inference instances from the train set and demonstrative examples from the test set. SST-2, RTE, CoLA, and MRPC are part of the GLUE benchmark (Wang et al., 2018), which does not provide labels for the test split. We require the labels for both demonstrative and inference examples. Thus, we use the validation set to select demonstrative examples for datasets part of GLUE and the train split to select examples that the model should classify.

Note that we apply our method on classification tasks, as this is a common task where ICL performs well (D'Oosterlinck et al., 2024; Rasheed et al., 2025; Cho and Inoue, 2025; Dong et al., 2024; Edwards and Camacho-Collados, 2024; Milios et al., 2023) and classification has a clear categorization, which we can then analyze for separability. In contrast, open-ended tasks like QA or generative tasks have a massive or even infinite answer space, complicating or prohibiting the analysis entirely.

4.2. Procedure

For a given task dataset, we first select our n demonstrative examples. The demonstrative examples are sampled randomly unless stated otherwise. These demonstrative examples remain the same for each run. We pass each inference instance appended to our ICL prompt through the model, extracting the hidden representations, and have the model respond. Then, we compute the accuracy using the responses given by the model. We are further interested in the separability (or lack thereof) shown by the generated hidden representations. Hence, we generate t-SNE visualizations of the embeddings as well as train a linear classifier to quantify the extent of separability.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

4.3. Models

We apply our procedure to seven models in total. Phi-4 (Abdin et al., 2024) with 14B parameters is our main model used for most experiments. We additionally conduct a subset of the experiments using Llama 2 (Touvron et al., 2023) with 7B and 13B parameters, Falcon 7B (Almazrouei et al., 2023), Llama 3 (Dubey et al., 2024) models with 3B and 8B parameters, and Qwen 2.5 (Yang et al., 2024) with 14B parameters. We use the instruction fine-tuned version and use default hyperparameter settings for all models.

5. Results

As mentioned previously, our primary hypothesis is that the ICL performance is governed by the LLM's ability to linearly discriminate between the instances of the constituent classes in the embedding space. This, in turn, is dependent on the demonstrative examples used in the prompt. We test this hypothesis first on the sentiment classification task (Section 5.1) and then generalize it to other text classification tasks (Section 5.2) and models (Section 5.3). Using insights into separability gained through these experiments, we further investigate peculiarities often observed in ICL (e.g., replacing ground truth labels with random ones does not degrade performance (Min et al., 2022)) through the lens of vector space geometry (Section 5.4) and apply it to select demonstrative examples (Section 5.5). All results are obtained using Phi-4 and four randomly selected demonstrative examples and 200 inference instances, unless specified otherwise.

5.1. Does separability influence classification accuracy?

The results presented here are for sentiment classification on the IMDb dataset. To gain an initial understanding of how the model processes the input, we aim to single out components that impact the performance and separability. In preliminary experiments, we observe that passing an ICL prompt, as described in Section 3, leads to clear separability, while passing only an inference instance, without instructions or examples, does not. This is intuitive in general and supports our separability hypothesis, as the demonstrative examples direct the model to better understand the objective of the task, formatting of the answers, and the label space.

Embeddings gained by a "default" run as described above are visualized in Figure 2a. Note that the embeddings created by the model during inference separate true positive and true negative instances well. Some outliers and false positive

instances are mixed in with true positives and negatives, but overall, the two classes are embedded somewhat distinctly. As issues can arise with t-SNE, we fit a linear classifier on the uncompressed embeddings to better gauge and confirm the separability. This linear classifier (LC) achieves an accuracy of 86.2% averaged across five "default" runs, suggesting good separability irrespective of t-SNE. We further observe that for our experiments, *changing the number of demonstrative examples has little effect on the t-SNE visualization and LC accuracy.* Moreover, *ordering has little impact on both classification accuracy (on average $< 5\%$ difference) and the separability*.

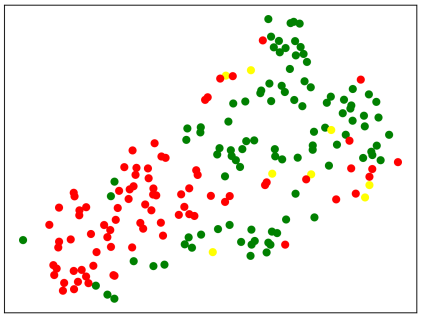
Discussion Our results indicate that LLMs can embed distinct classes separately. We see that demonstrative examples can improve (or in some cases harm) separability and that this separability is correlated with performance (Figure 1). It is therefore important to carefully consider which examples to use and not default to random sampling. Interestingly, the model generally creates similar embeddings for false positives as it does for true positives. This indicates that an instance the model deems "positive" has a certain internal representation, and the model decides, based on this representation, how to respond. It indicates that the *location of the embedding in the vector space is relevant* for the response to some extent. Note that internally this need not be encoded as the location in a vector space, but it can be a clearer illustration.

Takeaway Classification accuracy and separability are correlated. The model creates an internal representation that represents a class, and these representations are dependent on the demonstrative examples used. It follows that there are examples that lead to a high separability, while others do not.

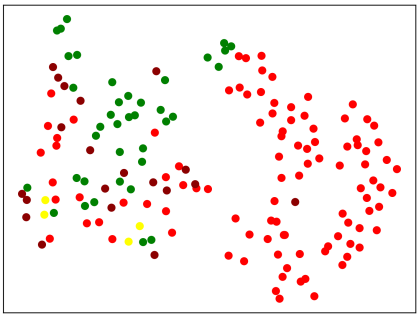
5.2. Transfer to Other Datasets

As mentioned in Section 3, sentiment analysis is found to be a task that is linearly separable in different setups. Although our findings above shine a new light on this task from a vector space perspective, we ask if the findings are only dependent on the ease of sentiment classification. To support the greater simplicity of sentiment analysis, we analyze the performance in a zero-shot setup, i.e., prompting the model without providing demonstrative examples.

We find that the accuracy remains rather high at around 72.7%. When training a linear classifier to predict the class of an embedding in the zero-shot setting, we achieve an accuracy of 72.0% (averaged across five runs), suggesting a lesser but still notable separability in higher dimensions. We further verify these results on SST-2, another sentiment analysis task. In the zero-shot setting,



(a) Default labels: "negative" / "positive"



(b) Replaced labels: "tsunami" / "husband"

Figure 2: t-SNE visualizations of runs with 200 samples, with 2 demonstrative examples per class and different label settings. The captions of the subfigures show the labels used for class 0 and class 1 in the ICL prompt. Figure 2a shows a "default" run, with no label changes. Figure 2b replaces the class labels by random, but fixed nouns, i.e., "positive" becomes "husband" and "negative" becomes "tsunami". **Key:** Green, red, yellow and dark red refer to true positive, true negative, false positive, and false negative, respectively.

the classification accuracy is around 86.1%. A linear classifier trained on the embeddings achieves an accuracy of 68.2%. With demonstrative examples, the classification accuracy increases slightly to 90.6% and the linear classifier score increases to 75.2%.

With these results, we can assume the generated embeddings for the sentiment are usually separable linearly, and more easily so when adding demonstrative examples. Yet, as we have a high accuracy and separability score in the zero-shot case, we also find that sentiment analysis is an easy task in general. To analyze whether this is applicable to other, more difficult tasks, we apply the same methodology to three new tasks, namely textual entailment (RTE), semantic equivalence (MRPC), and grammatical acceptability (CoLA).

Table 1 shows the classification accuracy and linear classifier accuracy for zero-shot and the default ICL prompt, with two demonstrative examples per class. We see that the zero-shot accuracy in both zero-shot and four-shot drops for RTE, MRPC and CoLA in comparison to both sentiment analysis tasks. Especially for RTE, the zero-shot classifica-

		IMDb	SST-2	RTE	MRPC	CoLA
0-shot	Acc.	72.7%	86.1%	5.0%	6.3%	17.0%
	LC	72.0%	68.2%	56.8%	61.4%	57.4%
4-shot	Acc.	93.6%	90.6%	87.7%	75.6%	81.2%
	LC	86.2%	75.2%	63.6%	66.0%	67.0%

Table 1: Classification accuracy (*Acc.*) and linear classifier (*LC*) accuracy for a zero-shot setting and ICL setting with two demonstrative examples per class (4-shot). The metrics are averaged over 5 runs. The demonstrative examples were selected randomly.

tion accuracy is 5% because close to all responses are out-of-space, e.g., textual descriptions of the task or further instructions not related to the answer. The improvement in classification accuracy by adding demonstrative examples is more significant for RTE, MRPC and CoLA. This is largely due to the better alignment to the correct formatting. The linear classifier accuracy increases as well, validating our hypothesis that linear separability is indeed correlated to ICL performance.

Discussion We find that sentiment analysis is an easy task. We have a high zero-shot accuracy and separability for IMDb and SST-2. For other tested tasks, we find a much poorer zero-shot performance across the board. The performance with no demonstrative examples is never above 20% accuracy, and the embeddings are poorly separable. Most generated responses in this case did not contain clear labels or no answers at all.

We can gather that sentiment analysis is thus a more straightforward task, as the instruction with no demonstrative examples is enough to generate mostly clear and distinct labels in the response. Adding demonstrative examples for the non-sentiment analysis datasets boosts the performance considerably, e.g., for RTE, an improvement of over 80 percentage points. Additionally, as the linear classifier scores improve, these results show that adding demonstrative examples not only helps models to learn how to respond to a prompt but also improves separability of embeddings generated by models, which leads to better performance.

Takeaway Sentiment analysis *is* an easy task that a model manages without ICL, but ICL can improve this good performance further and especially help the model learn to accomplish more difficult tasks. Moreover, ICL can help separability of embeddings, which can in turn enable more in-depth analysis.

5.3. Transfer to Other Models

The results above were all obtained with Phi-4. In-context learning has been found to be an emergent ability (Wei et al., 2022a), so we will analyze how our investigation holds for other models. Although 14 billion parameters is small compared to the number of parameters of commercial models like ChatGPT, there are smaller models that have between three and eight billion parameters. We run a more concise analysis on other models, namely Llama 2 7B, Falcon 7B, Llama 3.1 8B, Llama 3.2 3B, Llama 2 13B, and Qwen2.5 14B. For each model, we conduct a default run and fit a linear classifier on the embeddings to ensure separability.

The accuracy for IMDb can be seen in the upper half of Table 2. Especially, the linear classifier score for the three Llama models is rather low. This suggests that the embeddings are less expressive and cannot be easily separated using the linear classifier. The linear classifier may be misaligned, as mentioned above, leading to a low LC accuracy, even though the data is separable. Nevertheless, the accuracy is high for IMDb for the small Llama models, although not as high as Falcon 7B, Llama 2 13B, and Qwen 2.5. For the latter three, the separation is clearer, reflected in the higher linear classifier accuracy and classification accuracy. Moreover, the jump between zero-shot and 4-shot learning is larger for Falcon 7B and Llama 2 13B than for the other models. Interestingly, for Qwen 2.5, the zero-shot performance is good, and, primarily, the linear classifier score increases.

The lower half of Table 2 shows the classification and linear classifier accuracy for MRPC. As with Phi-4, MRPC is a more difficult task, leading to more misclassifications. Mostly, this can be attributed to the models generating non-fitting responses or adding further explanations instead of answering the prompt in the zero-shot case. While this is less the case in the 4-shot setting, we also have some false positives and false negatives, leading to relatively low scores. Compared to IMDb, the accuracy, both classification and linear classifier, is lower, as we saw with Phi-4. Nevertheless, we see the same increased separability for Falcon 7B, Llama 2 13B, and Qwen 2.5 when compared to the smaller Llama models.

Discussion In total, we find that small Llama models are much less separable. This suggests that the embeddings generated by these models are not expressive enough, i.e., in the case of IMDb, the embeddings do not encode enough information to separate sentiment as clearly.

As the embeddings are larger for Phi-4 and comparable models, and the models have more layers, the generated embeddings are more expressive than for small models. Furthermore, we see a good zero-shot separability for larger models. This separability is greatly improved when adding demonstrative examples. Some of these models, e.g., Falcon 7B, have a worse zero-shot classification accuracy than the smaller models. Yet, adding demonstrative examples for Falcon 7B, Llama 2 13B, and the

		Model	L2 7B	L3 8B	L3 3B	Falcon 7B	L2 13B	Qwen 2.5	Phi-4
IMDb	0-shot	Accuracy	82.5%	82.7%	72.2%	67.8%	85.7%	90.5%	72.7%
		LC	49.6%	52.2%	52.2%	78.0%	75.8%	76.6%	72.0%
	4-shot	Accuracy	90.0%	90.0%	90.8%	93.4%	91.2%	90.9%	93.6%
		LC	56.2%	54.2%	56.6%	86.0%	86.4%	88.6%	86.2%
MRPC	0-shot	Accuracy	26.5%	31.1%	19.9%	31.9%	6.8%	43.1%	6.3%
		LC	51.8%	51.6%	47.6%	58.0%	51.8%	63.2%	61.4%
	4-shot	Accuracy	66.8%	66.7%	60.7%	66.8%	72.1%	71.0%	75.6%
		LC	53.6%	53.8%	54.6%	65.6%	57.4%	68.8%	66.0%

Table 2: Accuracy and linear classifier score (LC) on IMDb and MRPC for various models. L2 and L3 refer to Llama 2 and Llama 3, respectively. For more model details, see Section 4.3.

14B models improves the classification accuracy to such an extent that they outperform the smaller models and have an increased separability. This means that models with good zero-shot separability greatly benefit, both in terms of separability and classification performance, from demonstrative examples.

Takeaway Small models' embeddings do not contain enough information to allow good separability. Large models create more expressive and separable embeddings with ICL, leading to a major performance increase.

5.4. Peculiarities in ICL

Prior work (Shi et al., 2024; Min et al., 2022) has highlighted some peculiarities in ICL, e.g., what impact changing the labels of demonstrative examples has on performance. As we have seen above, the performance is related to vector space separability. Consequently, we aim to use our vector space hypothesis to explain the observations of prior work. We consider four setups — (i) default, i.e., no changes to the labels; (ii) flipped labels, i.e., we flip the labels for the demonstrative examples (Shi et al. (2024)); (iii) blank labels, i.e., we insert blank labels ("_____") in place of the demonstrative example labels in the prompt. This removes any notion of what the possible classes are, and (iv) absurd labels, i.e., we replace the labels with a random but fixed noun[3] (Min et al. (2022)). This supplies labels that are not opposites (like negative vs. positive) to remove any inherent separation in the label phrasings. We apply our setups to sentiment classification.

In general, we expect the model to adhere to the class labels provided for the respective class. We refer to the original labels, i.e., class 0 is "negative" and class 1 is "positive", as ground truth labels. We calculate the accuracy w.r.t. the labels listed in Table 3 for each case, unless stated otherwise.

The default run shown in Figure 2a achieves an accuracy of 92%. To examine the dependency between the label and example text, in the second case (flipped labels), we use 200 inference instances as before; however, we set all occurrences of "positive" in the demonstrative examples to "negative" and vice versa. The accuracy w.r.t. the flipped labels is low at 6.5%. In contrast, the accuracy w.r.t. the non-flipped labels only drops slightly to 88.5%, indicating that simply flipping the labels in the prompt leads to no substantial internal change in the model to adhere to these "new" labels. The separation of ground truth positive and ground truth negative instances remains clear, as fitting a linear classifier on the embedding resulting from five such runs results in an LC accuracy of 85.8%, a slight decrease from the default run. This explains the marginal decline in performance despite flipping labels.

In the blank label scenario, the model only repeats the blank labels during inference, except in one case. Thus the measured accuracy is 99.5%. Regardless, the embeddings remain more-or-less separated by ground truth class, and the LC accuracy remains high at 79.2%.

Lastly, for the absurd label scenario, Figure 2b shows the embeddings when replacing the class label for "positive" with "husband" and replacing "negative" with "tsunami". Given this replacement, the output of the model largely adheres correctly to these new class labels. The accuracy calculated with "husband" and "tsunami" as class labels is

Case	Class 0	Class 1	Accuracy
(i)	"negative"	"positive"	92.0%
(ii)	"positive"	"negative"	6.5% (85.5%)
(iii)	"_____"	"_____"	99.5% (0.5%)
(iv)	"tsunami"	"husband"	81.0% (18.5%)

Table 3: Overview of labels used as class labels for demonstrative examples. We expect the model to adhere to these labels. We refer to the original labels (as in Case (i)) as ground truth labels. The accuracy listed is calculated w.r.t. the class labels. Where applicable, we report the accuracy w.r.t. ground truth labels in parentheses. The accuracies are achieved with Phi-4.

[3]The nouns are generated using wonderwords (https://wonderwords.readthedocs.io) with a length between 5 and 10 characters.

81.0%, while the LC accuracy is 82.0%. Out-of-space responses (i.e., not "husband" or "tsunami") are not shown in the visualization, but we note that the accuracy w.r.t. to the ground truth labels is 18.5%.

Discussion The above results indicate that *ICL can adapt to the provided labels to a considerable extent if the instances remain separable in the vector space*. We further find that inverting the labels does not hinder the split of embeddings w.r.t. to ground truth labels and does not significantly decrease the classification accuracy. Flipping the labels does not drastically change the vector space representation, so the model can still output the correct (ground truth) labels.

On the other hand, replacing the demonstrative example labels with blank labels results in the model generating useless responses. Following this, the model can generate roughly distinct embeddings for each class yet still comply with the formatting suggested (e.g., blank labels) in the demonstrative examples, even *overwriting prompt instructions* (e.g., "Classify as negative or positive"). We further see the importance of label space in the scenario when replacing the labels with random but fixed words. We see that the model can learn the "new" labels and correctly match these new labels to the examples' ground truth with only four demonstrative examples.

It is unclear what connection the model assigns between demonstrative examples and the specified labels. In the flipped labels case, there appears to be no connection, as the model does not adapt to the flipped classification labels, but when adding random, fixed words, the model does learn correctly what the new labels are. A likely explanation is that the model learned what "positive" and "negative" mean during pre-training and ignores the minor inconsistency in the demonstrative examples when flipping labels. This recovery of inconsistencies is similar to models fixing typos or unscrambling text in prompts (Penteado and Perez, 2023; Cao et al., 2023). In the case of replacing the labels with other words, the label space is drastically different from what the model learned, and the model "realizes" that no recovery is needed, but it needs to use different labels for each class.

Takeaway The model does not learn how to complete a task, like sentiment analysis, from the ground up by seeing demonstrative examples, as the classification performance on blank labels is poor, yet the embeddings can still be split by class. Rather, demonstrative examples help refine the formatting and label space and can even enable the model to completely relearn what labels correspond to the relevant class. We also see a link between demonstrative examples and their assigned label, although this link is not always precise.

Selection	IMDb	SST-2	RTE	MRPC	CoLA
Random	93.6%	90.6%	87.7%	75.6%	81.2%
Centroid	93.0%	91.9%	86.8%	73.8%	84.2%
Separated	$\mathbf{96.2\%}$	$\mathbf{92.8\%}$	$\mathbf{88.5\%}$	$\mathbf{78.2\%}$	$\mathbf{86.9\%}$

Table 4: Classification accuracy for different selection methods. We use embeddings created by Phi-4 to select demonstration examples. The accuracy is averaged over 5 runs with 200 inference instances per run.

5.5. Demonstrative Example Selection

We hypothesized that improving separability also improves classification accuracy and that ICL helps in clearly separating the embeddings. We have shown that this hypothesis holds, especially for larger models. Following these results, we should be able to select demonstrative examples with well-separated embeddings to improve classification performance. We run a small case study in which we select demonstration examples that, when passed through the model, create well-separated embeddings. This case study is only meant as a proof-of-concept, and we leave the implementation of a complete selection strategy using this method as future work.

We use Phi-4 to create embeddings of 1000 candidate examples per dataset per run. For each run, we then select two demonstration examples per class according to the selection strategy. First, random selection, which is the default in most studies. Second, centroid selection, which selects the "center-most" embeddings in our embedding space of 1000 examples. Lastly, separated selection selects the two examples per class such that the intra-class distance is minimized and the inter-class distance is maximized.

Table 4 presents the classification accuracy for Phi-4 across datasets for different selection strategies. We find that separated selection improves over random and centroid selection. Further, selecting the center-most examples performs worse than random selection on three of five datasets. This highlights the benefits of using well-separated examples, as centroid always has poorly separated examples, while random may select well-separated examples by chance.

6. Conclusion

In this work, we analyze in-context learning and shed some more light on why ICL works using a vector-space perspective. Our research shows that ICL helps models conform their output formatting and adapt their pre-existing knowledge rather than learning new tasks entirely. We show that LLMs create expressive embeddings when performing ICL for classification, such that these embeddings

can be separated by their class. Further, embeddings for difficult tasks (e.g., grammatical acceptability) are not as easily separable as those for easier tasks (e.g., sentiment analysis), paralleling classification performance on these tasks. The separability of generated embeddings can serve as a predictor of performance, and this separability can be improved through the selection of well-separated demonstrative examples, particularly in larger, more capable models. Generally, the larger the model, the more expressive the embeddings are, and small models, mostly, do not encode enough information to significantly split embeddings by their class. We believe our proposed selection method shows the advantage of looking at ICL from a vector space perspective for practical applications.

7. Acknowledgements

This work was partially funded by the Bundesministerium für Wirtschaft und Energie (BMWE), Germany, in the context of the 8ra Initiative ("Soofi", 13IPC040E).

8. Bibliographical References

Marah I Abdin, Jyoti Aneja, Harkirat S. Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *CoRR*, abs/2412.08905.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. 2023. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Lisa Alazraki, Maximilian Mozes, Jon Ander Campos, Tan Yi-Chern, Marek Rei, and Max Bartolo. 2025. No need for explanations: Llms can implicitly learn from mistakes in-context. *arXiv preprint arXiv:2502.08550*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

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

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8898–8913. Association for Computational Linguistics.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8123–8144. Association for Computational Linguistics.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. 2024. VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *CoRR*, abs/2406.05370.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen R. McKeown, and He He. 2023. On the relation between sensitivity and accuracy in in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 155–167. Association for Computational Linguistics.

Hakaze Cho and Naoya Inoue. 2025. Staicc: Standardized evaluation for classification task in in-context learning. *CoRR*, abs/2501.15708.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4005–4019. Association for Computational Linguistics.

Gilad Deutch, Nadav Magar, Tomer Bar Natan, and Guy Dar. 2024. In-context learning and gradient descent revisited. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1017–1028. Association for Computational Linguistics.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. 2025. Learning without training: The implicit dynamics of in-context learning. *CoRR*, abs/2507.16003.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1107–1128. Association for Computational Linguistics.

Karel D'Oosterlinck, Omar Khattab, François Remy, Thomas Demeester, Chris Develder, and Christopher Potts. 2024. In-context learning for extreme multi-label classification. *CoRR*, abs/2401.12178.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Aleksandra Edwards and José Camacho-Collados. 2024. Language models for text classification: Is in-context learning enough? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 10058–10072. ELRA and ICCL.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *CoRR*, abs/2209.10652.

Yanhe Fu, Yanan Cao, Qingyue Wang, and Yi Liu. 2024. TISE: A tripartite in-context selection method for event argument extraction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1801–1818. Association for Computational Linguistics.

Jun Gao, Ziqiang Cao, and Wenjie Li. 2024. Unifying demonstration selection and compression for in-context learning. *CoRR*, abs/2405.17062.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 14892–14904. Association for Computational Linguistics.

Shivanshu Gupta, Clemens Rosenbaum, and Ethan R. Elenberg. 2024. Gistscore: Learning better representations for in-context example selection with gist bottlenecks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Trans. Mach. Learn. Res.*, 2023.

Pradyoth Hegde, Santosh Kesiraju, Jan Svec, Simon Sedlácek, Bolaji Yusuf, Oldrich Plchot, Deepak K. T, and Jan Cernocký. 2025. Factors affecting the in-context learning abilities of llms for dialogue state tracking. *CoRR*, abs/2506.08753.

Oskar John Hollinsworth, Curt Tigges, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US. Association for Computational Linguistics.

Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. 2024. On the origins of linear representations in large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024a. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Jia Li, Yunfei Zhao, Yongmin Li, Ge Li, and Zhi Jin. 2023. Towards enhancing in-context learning for code generation. *CoRR*, abs/2303.17780.

Xiang Li, Haoran Tang, Siyu Chen, Ziwei Wang, Ryan Chen, and Marcin Abram. 2024b. Why does in-context learning fail sometimes? evaluating in-context learning on open and closed questions. *CoRR*, abs/2407.02028.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6219–6235. Association for Computational Linguistics.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. 2024. Uncertainty quantification for in-context learning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3357–3370. Association for Computational Linguistics.

Jiayu Liu, Zhenya Huang, Chaokun Wang, Xunpeng Huang, ChengXiang Zhai, and Enhong Chen. 2025. What makes in-context learning effective for mathematical reasoning. In *Forty-second International Conference on Machine Learning*.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. 2024. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Quanyu Long, Jianda Chen, Wenya Wang, and Sinno Jialin Pan. 2024a. Large language models know what makes exemplary contexts. *CoRR*, abs/2408.07505.

Quanyu Long, Yin Wu, Wenya Wang, and Sinno Jialin Pan. 2024b. Does in-context learning really learn? rethinking how large language models respond and solve tasks via in-context learning. In *First Conference on Language Modeling*.

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

May 22-27, 2022, pages 8086–8098. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. *CoRR*, abs/2309.10954.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Azza Mohamed, Mohamed El Rashid, and Khaled Shaalan. 2025. In-context learning in large language models (llms): Mechanisms, capabilities, and implications for advanced knowledge representation and reasoning. *IEEE Access*, 13:95574–95593.

Vinay M.S., Minh-Hao Van, and Xintao Wu. 2024. In-context learning demonstration selection via influence analysis. *CoRR*, abs/2402.11750.

Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. 2024. Learning vs retrieval: The role of in-context examples in regression with llms. *arXiv preprint arXiv:2409.04318*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2023, Singapore, December 7, 2023*, pages 16–30. Association for Computational Linguistics.

Maria Carolina Penteado and Fábio Perez. 2023. Evaluating GPT-3.5 and GPT-4 on grammatical error correction for brazilian portuguese. *CoRR*, abs/2306.15788.

Jian Qian, Miao Sun, Sifan Zhou, Ziyu Zhao, Ruizhi Hun, and Patrick Chiang. 2024. Sub-sa: Strengthen in-context learning via submodular selective annotation. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 2034–2041. IOS Press.

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

Areeg Fahad Rasheed, Safa F. Abbas, and M. Zarkoosh. 2025. Exploring in-context learning: A deep dive into model size, templates, and few-shot learning for text classification. In *Innovation and Emerging Trends in Computing and Information Technologies*, pages 207–215, Cham. Springer Nature Switzerland.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. 2024. Why larger language models do in-context learning differently? In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chenming Tang, Zhixiang Wang, and Yunfang Wu. 2024. Large language models might not care what you are saying: Prompt format beats descriptions. *CoRR*, abs/2408.08780.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy

Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023a. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.

Johannes von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. 2023b. Uncovering mesa-optimization algorithms in transformers. *CoRR*, abs/2309.05858.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6287–6310. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

Taihang Wang, Xiaoman Xu, Yimin Wang, and Ye Jiang. 2024. Instruction tuning vs. in-context learning: Revisiting large language models in few-shot computational social science. *CoRR*, abs/2409.14673.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for incontext learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguistics*, 7:625–641.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1423–1436. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang,

Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Bingsheng Yao, Guiming Chen, Ruishi Zou, Yuxuan Lu, Jiachen Li, Shao Zhang, Yisi Sang, Sijia Liu, James A. Hendler, and Dakuo Wang. 2024. More samples or more prompts? exploring effective few-shot in-context learning for llms with in-context sampling. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1772–1790. Association for Computational Linguistics.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024a. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2765–2781. Association for Computational Linguistics.

Zixiao Zhu, Zijian Feng, Hanzhang Zhou, Junlang Qian, and Kezhi Mao. 2024b. MICL: improving in-context learning through multiple-label words in demonstration. *CoRR*, abs/2406.10908.