Identifying and Analyzing Performance-Critical Tokens in Large Language Models

Yu Bai[1,2*], **Heyan Huang**[1,3], **Cesare Spinoso-Di Piano**[4,5], **Sanxing Chen**[6], **Marc-Antoine Rondeau**[4], **Yang Gao**[1†], **Jackie Chi Kit Cheung**[4,5,7]

[1] Beijing Institute of Technology, Beijing, China
[2] Beijing Academy of Artificial Intelligence, Beijing, China
[3] Southeast Academy of Information Technology, Fujian, China
[4] Mila - Quebec Artificial Intelligence Institute
[5] McGill University
[6] Duke University
[7] Canada CIFAR AI Chair

Abstract

In-context learning (ICL) has emerged as an effective solution for few-shot learning with large language models (LLMs). However, how LLMs leverage demonstrations to specify a task and learn a corresponding computational function through ICL is underexplored. Drawing from the way humans learn from content-label mappings in demonstrations, we categorize the tokens in an ICL prompt into content, stopword, and template tokens. Our goal is to identify the types of tokens whose representations directly influence LLM's performance, a property we refer to as being *performance-critical*. By ablating representations from the attention of the test example, we find that the representations of informative content tokens have less influence on performance compared to template and stopword tokens, which contrasts with the human attention to informative words. We give evidence that the representations of performance-critical tokens aggregate information from the content tokens. Moreover, we demonstrate experimentally that lexical meaning, repetition, and structural cues are the main distinguishing characteristics of these tokens. Our work sheds light on how LLMs learn to perform tasks from demonstrations and deepens our understanding of the roles different types of tokens play in LLMs.

1 Introduction

In-context learning (ICL) has become a popular technique employed with large language models (LLMs) (Brown et al. 2020). However, ICL has been shown to be unstable in that slight changes to the in-context prompts (e.g., reordering of demonstrations) can lead to substantial differences in performance (Lu et al. 2022; Zhang, Feng, and Tan 2022). This circumstance is difficult to control due to a lack of understanding of the model's working mechanisms, leaving us uncertain about the exact process by which LLMs learn to infer a task specification from demonstrations and produce a computation function to implement that task specification. Previous papers explored this issue, focusing on specific aspects such as the label space (Min et al. 2022) and the hidden states of the last prompt token (Hendel, Geva, and Globerson 2023; Todd et al. 2023), but have been limited in scope.

In this work, we conduct a comprehensive study on how LLMs extract information that is valuable for improving task performance from demonstrations. Drawing from the way humans learn through content-label mappings in demonstrations, we categorize the tokens in an ICL prompt into content, stopword (Wilbur and Sirotkin 1992), and template tokens, with the latter two typically ignored by humans due to their uninformative nature (Lenartowicz et al. 2014). We use content tokens as a category because they "are fixated (by human) about 85% of the time" (Rayner 1998). With these categories, we ablate the representations of different token types from the attention of ICL test examples, masking partial information during the model's task-solving process, as shown in Figure 1. This ablation is intended to identify the types of tokens whose representations LLMs **directly** depend on to achieve high-level performance, thereby explaining how LLMs learn from demonstrations. We refer to these tokens that are critical for performance as **performance-critical tokens**.

We show that template and stopword tokens (e.g., "Answer:") are the most prone to be performance-critical tokens. In contrast, content tokens (e.g., "Union", etc.) have a negligible impact on performance when their representations are eliminated from the attention of the test examples. This finding is counterintuitive since the original text of template and stopword tokens inherently do not possess any information found in the demonstrations. To explain this, we study the relationship among different types of tokens through ablation experiments that cut off the information flow between different kinds of tokens. We show that content tokens are **indirectly** leveraged by LLMs during ICL through aggregating their information into the representations of performance-critical tokens, raising questions about how LLMs use these

[*] Work done during the internship at Mila. The full paper, including all technical appendices, is available at *arXiv:2401.11323*.
[†] Corresponding author.

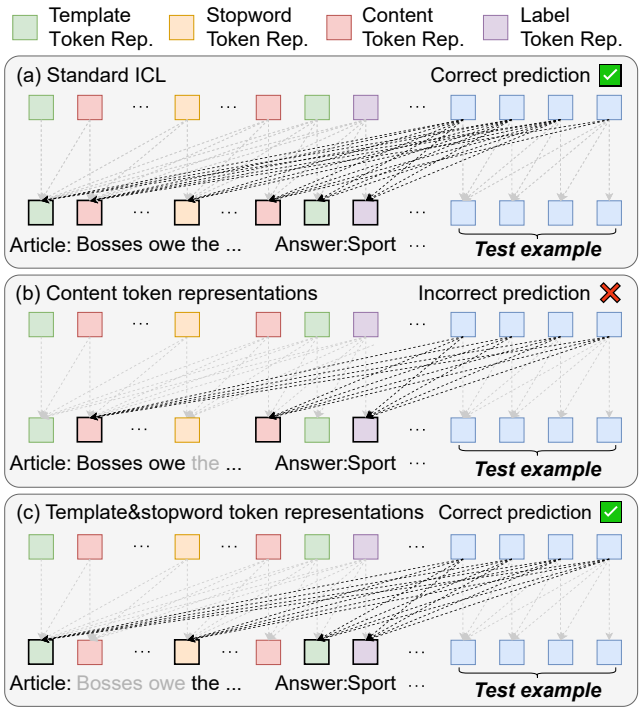

Figure 1: An illustration of 4-way text classification on AGNews with different parts of its 4-shot ICL demonstrations masked with respect to the attention of the test example. Specifically, (b) masking the representations of what we call the template and stopword tokens from the attention of the test example leads to a significant drop in performance while (c) masking representations of the content tokens leaves the performance relatively unchanged. The dashed lines represent the attention between every pair of tokens while those from the test example to the ICL prompt are unshaded. "Rep." stands for "representation".

tokens in general text processing and reasoning.

Beyond identifying performance-critical tokens, we analyze them to better understand how they are leveraged by LLMs. Specifically, we investigate the characteristics which differentiate them from other tokens. We find the following three distinguishing characteristics: the **lexical meaning** of tokens as it relates to the task being solved, the **repetition** of tokens throughout the prompt, and the **structural cues** which the tokens provide to the prompt. Our findings indicate that the lexical meaning, repetition, and structural cues of performance-critical tokens contribute to task performance across all model sizes, suggesting that they contribute to the property of tokens being performance-critical.

Our work reveals that we can identify and characterize the types of tokens whose representations are the most important in directly maintaining ICL task performance. Our results suggest that previous claims about ICL should be more nuanced, in that representations of tokens beyond label words (Wang et al. 2023) may also directly impact the task performance. We investigate the characteristics of lexical meaning, repetition, and structural cues related to performance-critical tokens which allow us to partially explain their importance and help us better understand how to avoid performance instability while using ICL. Additionally, our findings reinforce that we cannot assume LLMs and humans solve tasks in similar ways, emphasizing the need to develop better explanations of LLM behaviors. Overall, our research deepens the understanding of the roles different types of tokens play in LLMs, pointing to future work that explores how specific token representations can be leveraged for particular purposes (e.g., storing compressed information). Code and data are released at https://github.com/ybai-nlp/PCT_ICL.

2 Related Work

Working mechanisms of ICL Since the proposal of in-context learning (Brown et al. 2020), its working mechanisms have been extensively studied by the research community (Min et al. 2022; Liu et al. 2021; Bhattamishra et al. 2023; Zhou et al. 2023). Min et al. (2022) suggest that demonstrations primarily provide the label space, the distribution of the input text, and the format of the sequence for the test example. They argue that the precise ground truth labels do not have significant importance. In contrast, Yoo et al. (2022) propose a differing view, stating that the impact of the ground truth labels depends on the experimental configuration. Mao et al. (2024) analyze in-context learning from the perspective of data generation. The perspective of supported training data is also leveraged to analyze ICL (Han et al. 2023). Zhao et al. (2024) propose to use coordinate systems to understand the working mechanism of in-context learning. Zhou et al. (2023) propose a comprehensive survey on the interpretation and analysis of in-context learning. Lin and Lee (2024) model the dual modes of in-context learning (task retrieval vs. task learning) and phenomena like early performance drops. Li et al. (2024) introduce a long-context benchmark revealing degradation with very long demonstrations.

Our work investigates the working of ICL in LLMs at inference time, demonstrating that certain specific tokens are more likely to possess representations that could affect the processing of final test samples, improving the performance.

Function vectors of in-context learning Todd et al. (2023) and Hendel, Geva, and Globerson (2023) provide evidence that function vectors store information used to solve a task in ICL. They probe and extract the hidden representations of the final tokens in the prompt, which can then be added to or replace the corresponding vectors in a zero-shot example, yielding results comparable to using all demonstrations as context. Liu, Xing, and Zou (2023) propose using an in-context vector to represent the target task and apply feature shifting to query examples. They feed each input and its target into an LLM, concatenate the latent states, and apply PCA to derive a vector aligned with the task. Finally, Wang et al. (2023) propose that label words in demonstrations act as information anchors, aggregating information from previous examples. This suggests label tokens may satisfy our definition of performance-critical tokens. Yang et al. (2025) show that compact task vectors naturally emerge and can be enhanced with an auxiliary loss. Tikhonov, Oseledets, and Tutubalina (2025) find that multiple vectors are often needed

Notation	Component example
$\mathbf{I}$	Classify the news articles into the categories of World, Sports, Business, and Technology.\n\n
$\mathbf{T}^{\text{in}}$	Article: $\{\mathbf{D}^{\text{in}}\}$\n
$\mathbf{T}^{\text{out}}$	Answer: $\{\mathbf{D}^{\text{out}}\}$\n\n
$\mathbf{D}_1^{\text{in}}$	Radio veteran Karmazin joins Sirius. Sirius Satellite Radio Inc. named former Viacom Inc. president Mel...
$\mathbf{D}_1^{\text{out}}$	Business
$\mathbf{D}_2^{\text{in}}$	Numbers point to NY. NEW YORK - The New York Yankees can achieve two milestones with one more...
$\mathbf{D}_2^{\text{out}}$	Sports
ICL Prompt	Classify the news articles into the categories of World, Sports, Business, and Technology. Article: Radio veteran Karmazin joins Sirius. Sirius Satellite Radio Inc. named former Viacom Inc. president Mel... Answer: Business Article: Numbers point to NY. NEW YORK - The New York Yankees can achieve two milestones with one more... Answer: Sports

Table 1: An example of the components of a 2-shot ICL prompt in the AGNews dataset.

for complex tasks. Dong et al. (2025) propose the Linear Combination Conjecture, explaining vector formation but noting limitations for high-rank tasks. Jiang et al. (2025) introduce function vectors to mitigate catastrophic forgetting in continual instruction tuning.

All these previous studies either solely focus on a single token (i.e., the last prediction prompt token or label token) of the ICL prompt or treat the entire demonstration as a single unit, neglecting the other tokens within it. Our research focuses on all the tokens in the prompt and reveals that there are additional tokens with specific characteristics whose representations significantly affect the final ICL performance.

3 Preliminaries

Notation In in-context learning (ICL), task demonstrations (e.g., input-output pairs) are leveraged to construct a structured prompt that guides the model in predicting the final answer. Formally, the structural prompt consists of the following components: the instruction $\mathbf{I}$, the templates $\mathbf{T}^{\text{in}}$, $\mathbf{T}^{\text{out}}$, and the demonstrations $\mathbf{D}_i^{\text{in}}$, $\mathbf{D}_i^{\text{out}}$, where i denotes the i^{th} demonstration while in and out refer to the input text and output labels, respectively. These prompt components are concatenated to form the ICL prompt, P, as shown in Table 1. During inference, the templated version of the test example without its answer, $\mathbf{T}^{\text{in}} \cdot \mathbf{D}_{\text{test}}^{\text{in}} \cdot \mathbf{T}^{\text{out}}$, is appended to the ICL prompt and then sent to the large language model to predict the corresponding answer. We use $\cdot$ to denote the concatenation of token sequences.

Definition of performance-critical token In defining the performance-critical tokens, we measure the performance variation before and after incorporating the representations of specific tokens into the attention scope of the test example. Specifically, we define performance-critical tokens as the tokens that lead to both a noticeable performance improvement when their representations are included in the attention of test examples and a noticeable performance degradation when they are excluded.

Formally, let M be a decoder-only transformer-based large language model (LLM) and D be a classification dataset. We define H_P as the set of representations of each token in the ICL prompt P and H_{test} as the set of representations of the test demonstration which is appended to P for prediction (i.e., $\mathbf{T}^{\text{in}} \cdot \mathbf{D}_{\text{test}}^{\text{in}} \cdot \mathbf{T}^{\text{out}}$). In addition, we let $H_{\text{attend}} \subseteq H_P$ be some set of representations which M may attend to from H_{test} at inference time while performing ICL. For instance, $H_{\text{attend}} := H_{\mathbf{I}}$ would imply that, when M is predicting the label of the test demonstration, the attention from the test example is restricted to the prompt's instruction token representations.

Then, in order to provide a practical definition for the performance-critical tokens, we let $\mathbf{Acc}(M, D, H_{\text{attend}})$ be the accuracy achieved by a LLM M when performing ICL on the classification dataset D where the only representations which the test example may attend to at inference time are H_{attend}. Given a partition $\mathcal{P}$ of H_{P}, we say that a set of tokens $H^* \in \mathcal{P}$ is *performance-critical* if

$$\begin{aligned} &\mathbf{Acc}(M, D, H^*) \gg \mathbf{Acc}(M, D, \emptyset) \quad \text{and} \\ &\mathbf{Acc}(M, D, H_P) \gg \mathbf{Acc}(M, D, H_P - H^*) \end{aligned} \tag{1}$$

We note that examining the possibility of each token being performance-critical (i.e., $|H^*| = 1$) in an ICL prompt would be computationally intractable. We instead categorize all the tokens based on the role they play in the prompt and identify which types of tokens are more likely to be performance-critical in Section 5.

4 Experimental Settings

Datasets. We consider the most widely used text classification datasets used by previous studies (Zhao et al. 2021). For topic classification, we use the 4-way and 14-way datasets AGNews and DBPedia (Zhang, Zhao, and LeCun 2015). We also use SST2 (Socher et al. 2013) and TREC (Voorhees and Tice 2000) for sentiment and question classification tasks. For textual entailment, we use the 3-way CB (De Marneffe, Simons, and Tonhauser 2019) and 2-way RTE dataset (Dagan, Glickman, and Magnini 2005). Results for these two datasets are shown in Appendix G. Besides these classification tasks, we also present results in **machine translation** and **question answering** tasks to show that our findings can also be extended to text generation tasks. Results and analyses of these generation tasks are attached to Appendix M.

Evaluation. For each dataset, we randomly select 4 training demonstrations from the training set using 15 seeds, limited by the computational cost of LLM inference. For testing, we evaluate each setting on 500 randomly selected test examples. We show this sample size is sufficient by comparing results with 500 test examples and the full dataset using OpenLlama 3B and Llama 7B models, as shown in Appendix K. Instruction prompt $\mathbf{I}$ is retained in all ablations, as it is essential for model performance (Yin et al. 2023). We use a fixed $\mathbf{I}$ for all main results of different models, with additional experiments with different $\mathbf{I}$ provided in Appendix L, demonstrating that changing $\mathbf{I}$ does not affect our main findings.

LLMs. We utilize the 7B, 13B, and 33B Llama models (Touvron et al. 2023a) and a 3B OpenLlama model. We also report additional results with Llama 2 7B, Llama 2

13B (Touvron et al. 2023b), Mistral 7B (Jiang et al. 2023), and Gemma 3 4B (Team et al. 2025) models. Models after supervised fine-tuning are tested in Appendix H. All experiments are conducted on a single A100 80G GPU. For the 13B and 33B models, we apply 8-bit quantization to fit them into a single GPU.

5 Performance-Critical Token Detection

In this section, we aim to identify the performance-critical tokens in the ICL prompt. We first structurally categorize all the tokens in the prompt into three types: template, stopword, and content tokens. Then, we provide supporting evidence from the view of task performance to show that the template and stopword tokens are the most prone to be performance-critical tokens. Finally, we demonstrate that the information of content tokens serve to indirectly contribute to the performance by being propagated into the representations of the performance-critical tokens by LLMs.

5.1 Token types

We categorize ICL tokens based on the structure of the ICL prompt, following our notation in Table 1. Firstly, we find it natural to categorize tokens based on the structure of ICL prompts where the tokens from the demonstration examples $\mathbf{D}^{\text{in}}$ and the labels $\mathbf{D}^{\text{out}}$ are separated by template tokens from $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$. Second, $\mathbf{D}^{\text{in}}$ can be subdivided into content and stopword tokens, with the latter typically providing less useful information and often being ignored (Rayner 1998) when humans use analogy to learn specific tasks. Guided by these intuitions, we categorize all the tokens in the ICL prompt into template tokens, stopword tokens, and content tokens as follows:

Template tokens (TEMP): In defining template tokens, we include all the tokens which serve as templates for the ICL prompt. This includes the tokens in $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$.

Stopword tokens (STOP): In defining stopword tokens, we include punctuation and conjunction words, such as [,], [.], etc., in the prompt. We use the stopword tokens appearing in the instructions[1].

Content tokens (CONT): In defining content tokens, we include all the tokens from $\mathbf{D}^{\text{in}}$ except for the ones that are already stopword tokens. We use the term "content tokens" as they convey the meaningful information found in the demonstrations.

Researchers might typically expect content tokens to be critical, as they account for most of the information in demonstrations. However, in our experiments, we find that the representations of template and stopword tokens have the greatest direct impact on performance.

5.2 Ablation on token types

To determine which token types are more likely to be performance-critical tokens whose representations directly affect the final performance significantly, we design two experiments which ablate representations or tokens based on token types. The first involves keeping and masking representations of different token types from the attention of the test example. The second involves dropping the various kinds of tokens from the ICL prompt. The main purpose of the first experiment is to identify the performance-critical tokens defined in Section 3, while the second experiment aims to cut off the information propagation of different types of tokens to further explore the functioning of performance-critical tokens. Illustrations of these two methods which we refer to as representation-level and token-level ablations are shown in Figure 1 and Figure 2. Examples for the representation-level ablation is provided in Appendix F.

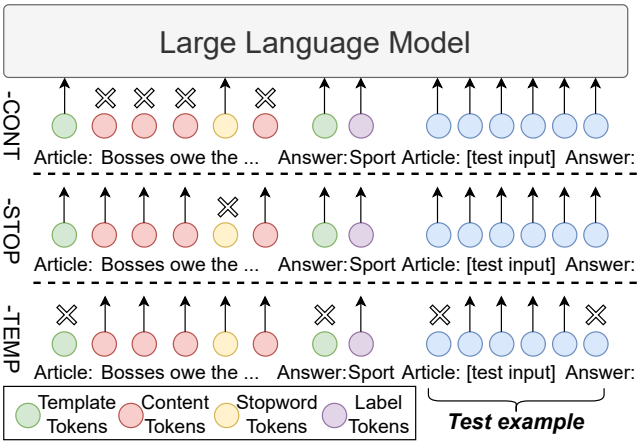

Figure 2: An illustrative example of the token-level ablation we use to analyze the working mechanism of performance-critical tokens.

Representation-level ablation Our first ablation stems from the intuition that if LLMs rely on the representations of specific token types to achieve high-level performance, they should perform adequately with only these representations. Performance should drop significantly if these tokens are removed from the test example's attention. Hence, we first pass the entire ICL prompt to the LLM and then restrict the test example's attention to representations of a particular token type (or types)[2] during its solving of the task. We compute task performances with every possible ablation combination, removing the representations of one (e.g., Standard ICL − TEMP) or two token types (e.g., Zero-shot + CONT[3]) from the attention of the test example. All the task performances and the averaged relative performance changes are reported, shown in Table 2, 3, and 4. Results for Llama 7B and 13B models are shown in Appendix G. An illustration of this set of experiments is shown in Figure 1.

Overall, these results demonstrate that **template and stopword tokens** are more likely to be performance-critical tokens than content tokens, conforming to our definition in

[1]The stopword token list used in the main experiments and the ablation results with the complete NLTK (Loper and Bird 2002) stopwords list are shown in Appendix I.

[2]Since $\mathbf{D}^{\text{out}}$ tokens have been shown to significantly impact performance (Wang et al. 2023), we always preserve the attention on the representations of the $\mathbf{D}^{\text{out}}$ tokens.

[3]Removing two token types from Standard ICL is equivalent to adding the other type to Zero-shot.

Models	Setting	AGNews	SST2	TREC	DBPedia	△Avg.
Llama 2 7B	Zero-shot	50.2	50.4	57.2	6.4	36.0
	+ CONT	0.9	61.0	50.6	12.9	+1.9
	+ STOP	49.0	78.1	54.4	61.6	+23.4
	+ TEMP	**81.1**	**82.6**	**55.2**	**65.5**	**+31.3**
Llama 2 13B	Zero-shot	56.2	90.8	49.0	7.6	53.3
	+ CONT	0.5	56.0	61.4	0.0	−14.1
	+ STOP	47.2	76.8	**65.2**	65.3	+9.6
	+ TEMP	**78.2**	**93.7**	62.4	**70.4**	**+18.1**
Mistral 7B	Zero-shot	77.8	84.4	73.0	57.8	59.5
	+ CONT	43.3	52.0	66.6	10.1	−10.0
	+ STOP	78.9	92.5	**71.6**	81.4	+18.4
	+ TEMP	**81.7**	**95.9**	63.9	**83.3**	**+19.7**
Llama 2 7B	Standard ICL	85.0	93.2	58.3	66.7	70.7
	− CONT	82.4	85.5	54.3	64.2	−3.8
	− STOP	84.8	88.0	51.7	65.7	−2.5
	− TEMP	0.9	61.0	50.6	12.9	−32.8
Llama 2 13B	Standard ICL	82.8	94.9	62.8	74.6	73.6
	− CONT	79.0	94.1	62.7	72.4	−1.3
	− STOP	80.1	89.4	61.5	74.1	−2.9
	− TEMP	0.5	56.0	61.4	0.0	−33.5
Mistral 7B	Standard ICL	82.2	97.0	67.4	82.4	80.2
	− CONT	81.8	96.2	64.4	83.4	−0.6
	− STOP	81.3	97.0	66.5	80.5	−0.8
	− TEMP	78.6	52.0	66.6	10.1	−22.8

Table 2: The accuracy results of the representation-level ablation study using Llama 2 and Mistral models where, for example, +TEMP refers to allowing attention only to template tokens and −TEMP refers to allowing attention only to content and stopword tokens. All values are presented as percentages. Results are averaged over 5 different random seeds. The best results are in bold and the results showing the greatest decrease during the ablation are underlined.

Models	Setting	AGNews	SST2	TREC	DBPedia	△Avg.
Open Llama 3B	Zero-shot	22.0	20.0	23.6	5.4	19.5
	+ CONT	26.2	52.1	30.1	7.4	+14.8
	+ STOP	36.7	82.9	**32.0***	52.4	+33.7
	+ TEMP	**56.5**	**86.7**	27.1	**62.2**	**+37.4**
Llama 33B	Zero-shot	70.2	88.6	60.6	30.2	54.6
	+ CONT	24.4	61.7	62.1	10.5	−6.7
	+ STOP	72.9	92.7	**66.7***	69.1	+17.7
	+ TEMP	**80.5**	**95.2**	65.2	**75.2**	**+24.6**
Open Llama 3B	Standard ICL	63.7	91.2	21.9	61.9	58.0
	− CONT	58.2	86.9	27.6	61.9	−0.9
	− STOP	51.8	78.9	28.8	30.3	−9.9
	− TEMP	26.2	52.1	30.1	7.4	−23.8
Llama 33B	Standard ICL	85.0	96.5	68.1	78.4	81.6
	− CONT	82.3	95.4	64.9	76.1	−1.5
	− STOP	84.8	94.9	62.1	77.3	−4.3
	− TEMP	24.4	61.7	60.6	10.5	−32.7

Table 3: The accuracy results of the representation-level ablation study using Llama models where, for example, +TEMP refers to allowing attention only to template tokens and −TEMP refers to allowing attention only to content and stopword tokens. All values are presented as percentages. Except where noted with *, all test statistics reported correspond to p-values < 0.05. The best results are in bold.

Setting	AGNews	SST2	DBPedia	△Avg.
Zero-shot	-	-	-	-
+ CONT	1.62	50.08	2.24	17.98
+ STOP	77.32	87.22	76.72	80.42
+ TEMP	**83.66**	**92.74**	**80.50**	**85.63**
Standard ICL	-	-	-	-
− CONT	83.66	92.74	80.50	85.63
− STOP	84.22	94.14	79.50	85.95
− TEMP	76.82	90.78	76.56	81.39

Table 4: The accuracy results of the representation-level ablation study using Gemma-3 4B models. Values are presented as percentages. Results are averaged over 5 random seeds.

Section 3. On the one hand, template token representations directly influence the task performance in ICL, achieving an average performance 39.8% higher than the zero-shot baseline by only utilizing these representations at inference time. If the representations of stopword tokens are further included (i.e., Standard ICL−CONT), the performance is nearly equivalent to that of the Standard ICL. In contrast, content token representations only bring an average improvement of 10.7%. On the other hand, the performance decreases the most with Standard ICL−TEMP, highlighting the direct significance of template tokens again[4]. Furthermore, considering token counts for each type, as shown in Appendix O, content tokens vastly outnumber the other two types. Hence, the averaged impacts of the template and stopword tokens further suggest that they are more prone to be performance-critical.

Rare exceptional cases appear when performance is relatively poor with Standard ICL (e.g., OpenLlama 3B in TREC). In some cases, masking the representations of the content tokens brings even better performance than the Standard ICL method, which is possibly due to the elimination of noisy information in the demonstration content. Another interesting observation is that the performance results of Standard ICL−STOP and Standard ICL−CONT where the attention to the content and stopword tokens is ablated respectively are similar, with an average difference of only 5.4%. This indicates that the representation of stopword tokens may contain overlapping information with their preceding content tokens. We believe that this could enable LLMs to model long sequences without significant architectural changes (e.g., using stopword representations as synthesis checkpoints) and leave the verification of this hypothesis to future work.

Token-level ablation In this section, we modify the ICL prompt by removing certain types of tokens from the ICL prompt[5] to further investigate the relationship between different kinds of tokens, cutting off the information flow between the representations of different tokens, shown in Figure 2. When we ablate the template tokens, we preserve the answer and next-line tokens in the templates to maintain a basic separator between the demonstration inputs and outputs. Results averaged on all the datasets are presented in Figure 3. Detailed results on each dataset could be seen in Appendix P.

[4]Both STOP and TEMP include the "\n" token; we mask the attention to the "\n" token as long as one of them is ablated in this set of experiments. Analyses about this experimental setting are shown in Appendix J.

[5]For template tokens, this includes *both* the tokens in the demonstrations and the test example to maintain their consistency. We included the analyses of only ablating the tokens in the demonstrations in Appendix Q.

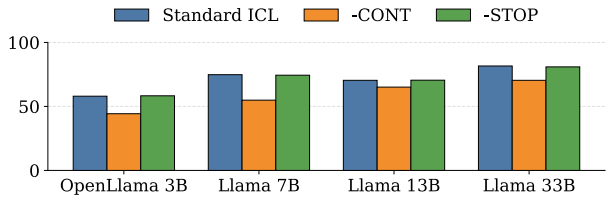

Figure 3: Results of the token-level ablation where, for example, −STOP refers to the ablation where stopword tokens are dropped from the ICL prompt. Models **without template tokens** consistently yielded **an accuracy of 0%** and are thus omitted from this figure.

Ablation(− CONT)	OpenLlama 3B	Llama 7B	Llama 13B	Llama 33B
Rep. level	57.1	72.8	70.6	80.1
Token level	44.3	54.9	65.1	70.4
Δ Difference	12.8	17.9	5.5	9.7

Table 5: The comparison between the −CONT performance of different models across two levels of ablations. "Rep. level" refers to representation level.

Our first finding from this ablation is that information is propagated from content token representations to performance-critical token representations, as shown by the contrast between representation-level and token-level ablation in Table 5. The representations of template and stopword tokens alone (i.e., Standard ICL − CONT in Figure 3) are less effective at affecting task performance, leading to worse performance than those where content information is included in their attention (i.e., Standard ICL − CONT in Table 3).

This finding provide us with additional insights about how LLMs leverage different kinds of tokens during ICL. Firstly, this circumstance means that even though the representations of the content tokens are not directly used when LLMs predict the answer, the encoding of these tokens contribute to the final performance indirectly through being aggregated into the representations of the performance-critical tokens. Secondly, it also suggests that LLMs prefer to utilize the the performance-critical tokens to aggregate the indirect information from the demonstration rather than others (i.e., content tokens). It is their incorporation of this information that makes them better at encoding tasks, partially explaining the working mechanism of ICL.

In addition, removing template tokens causes the LLMs to completely lose their ability to solve tasks via ICL with an overall task accuracy performance of 0% for all sizes and all tasks. We hypothesize that this is because the model no longer has an explicit cue to generate the target label, which is further discussed in Section 6. In this case, if we add back the last prompt token after the next-line token, the results return to their original level due to the introduction of a template token. This finding confirms previous claims that the format of ICL prompts plays a significant role in retaining performance (Min et al. 2022).

Roles of different types of tokens. To summarize, template and stopword tokens are the most performance-critical. Template token representations significantly improve the task performance, while the representations of stopword tokens play a more supportive role in the spectrum of performance-critical tokens by summarizing the information of content tokens. In contrast, content token representations do not **directly** contribute to the performance but instead **indirectly** provide information that is aggregated into the other two types of tokens. We discuss possible applications of these findings in Appendix R. This finding further raises an additional question: What are the characteristics for a token to be perceived by a LLM as performance-critical?

Models	Settings	AGNews	SST2	TREC	DBPedia	Avg.
Open Llama 3B	Standard ICL	63.7	91.2	21.9	61.9	58.0
	Swap	64.4	86.8	21.7	58.7	57.8
	$\text{Random}_{\text{fixed}}$	57.5	71.4	32.4	51.2	52.6
Llama 7B	Standard ICL	82.4	94.3	63.5	68.7	74.8
	Swap	70.2	11.4	44.3	58.2	49.8
	$\text{Random}_{\text{fixed}}$	19.5	11.4	13.2	7.4	15.5
Llama 13B	Standard ICL	81.6	94.3	60.0	76.1	70.4
	Swap	81.5	67.4	36.4	75.9	63.7
	$\text{Random}_{\text{fixed}}$	52.1	76.8	27.7	48.9	49.3
Llama 33B	Standard ICL	85.0	96.5	68.1	78.4	81.6
	Swap	84.5	94.9	60.8	75.5	73.2
	$\text{Random}_{\text{fixed}}$	78.7	92.5	52.2	75.8	68.2

Table 6: Results validating the effect of lexical meanings of template tokens, presented as percentages. The results showing the greatest decrease during the disruption are underlined.

6 Performance-Critical Token Analyses

To answer the above question, we provide analyses of the tokens whose representations we believe mainly store information that directly affects the performance of a task drastically. We focus on the template tokens since, as evidenced by the findings in Section 5.2, **their representations are the most important to maintaining task performance**. Our analysis focuses on the distinguishing characteristics of performance-critical tokens, while we also examine the effects of each part of template tokens in Appendix D.

By better understanding what characteristics of performance-critical tokens lead them to affect task performance, we provide insights on how to best leverage LLMs for ICL (e.g., What principles should practitioners be using when designing prompt templates?). We hypothesize that the following characteristics are critical for a token to be leveraged as performance-critical tokens: **lexical meaning** referring to the task-related lexical meaning of a performance-critical token, **repetition** referring to the multiple appearances of the performance-critical tokens in the prompt, and **structural cues** referring to how performance-critical tokens format the ICL prompt, shown in Table 1, into structured text.

We design several experiments to test if these characteristics affect the impact of performance-critical tokens, by disrupting each characteristic in the ICL prompts. A characteristic is related if there is a performance drop after the disruption. The disruption is achieved by replacing the template tokens with different kinds of random string templates. We use 5 different random string templates which are at-

Models	Settings	AGNews	SST2	TREC	DBPedia	Avg.
OpenLlama 3B	$\text{Random}_{\text{fixed}}$	**57.5**	**71.4**	**32.4**	**51.2**	**52.6**
	$\text{Random}_{\text{nonfixed}}$	30.2	71.4	17.1	18.6	38.8
Llama 7B	$\text{Random}_{\text{fixed}}$	**19.5**	11.4	**13.2**	**7.4**	**15.5**
	$\text{Random}_{\text{nonfixed}}$	15.5	**11.6**	10.4	1.8	11.6
Llama 13B	$\text{Random}_{\text{fixed}}$	**52.1**	**76.8**	**27.7**	**48.9**	**49.3**
	$\text{Random}_{\text{nonfixed}}$	32.1	34.5	19.2	6.0	24.3
Llama 33B	$\text{Random}_{\text{fixed}}$	**78.7**	**92.5**	**52.2**	**75.8**	**68.2**
	$\text{Random}_{\text{nonfixed}}$	78.5	87.5	46.3	63.1	64.2

Table 7: Results validating the effect of repetitive patterns, presented as percentages. We bold the highest accuracy for each classification task and model size.

Models	Settings	AGNews	SST2	TREC	DBPedia	Avg.
OpenLlama 3B	Standard ICL	70.7	51.7	40.4	53.5	53.3
	$\text{Random}_{\text{fixed}}$	47.5	51.8	32.6	19.4	40.9
Llama 7B	Standard ICL	72.3	77.4	54.1	64.7	64.3
	$\text{Random}_{\text{fixed}}$	3.9	16.9	3.5	9.6	10.2
Llama 13B	Standard ICL	82.0	72.0	60.1	75.9	70.1
	$\text{Random}_{\text{fixed}}$	46.1	47.5	25.0	50.8	39.7
Llama 33B	Standard ICL	85.3	88.3	71.2	75.5	76.9
	$\text{Random}_{\text{fixed}}$	69.7	53.0	37.8	72.8	54.0

Table 8: One-shot experimental results validating the effect of structural cues, presented as percentages. Models **without template tokens** consistently yielded **an accuracy of 0%** and are thus omitted from this table.

tached to Appendix U and average all the results for each setting.

Lexical meaning. A performance-critical token might be more impactful on the performance based on its lexical meaning. One hypothesis is that if the token carries task-related meanings like "Answer", it is more likely to serve as a performance-critical token.

To verify if lexical meanings could affect the formation of performance-critical tokens, we 1) Replace the tokens from $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ with the same random strings across the different demonstrations (**Random**$_{\text{fixed}}$), thus completely disrupting the lexical characteristic of these tokens; 2) Swap $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ (**Swap**), thus partially disrupting the lexical characteristic of these tokens. Table 6 shows that disrupting the lexical meaning of tokens slightly impacts task performance in smaller models (OpenLlama 3B), while larger models experience more significant drops. Llama 7B, in particular, is highly sensitive to lexical meaning and performs worse when semantics are disturbed. Thus, the lexical meaning of tokens likely influences their performance-critical nature.

Repetition. The impact of performance-critical tokens could also be influenced by their repetition throughout the prompt. Intuitively, via the attention mechanism, repetitive patterns are more likely to propagate information through the processing of text. Yan et al. (2024) propose self-reinforcement in in-context learning, also suggesting that repetition could be a significant factor in ICL.

We explore the repetition characteristic by comparing the results of the previously discussed **Random**$_{\text{fixed}}$ experiment with an experiment replacing $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$ with different random strings (**Random**$_{\text{nonfixed}}$), thus breaking the repetition of template tokens present in ICL demonstrations. We further conduct experiments **using template tokens with specific lexical meanings for comparison**, as detailed in Appendix V. Table 7 shows that without consistent repetition of performance-critical tokens, performance decreases for most models. This suggests that necessary information may not have been properly accumulated in the template token representations. These experiments demonstrate that repetition significantly influences the impact of performance-critical tokens. The results align with previous findings, reinforcing our claim that repetition is a key characteristic.

Structural cues. Beyond lexical meaning and repetition, the influence of performance-critical tokens may also depend on how ICL prompts are formatted. ICL prompts often include structural cues to assist the model to differentiate between elements with distinct roles, such as task inputs and target labels, within a demonstration. For example, template tokens (i.e., $\mathbf{T}^{\text{in}}$ and $\mathbf{T}^{\text{out}}$) delimit demonstration examples and labels, while stopword tokens (e.g., ,", ."., ":", etc.) structure content words into sentence components. Examples of how performance-critical tokens delimit ICL prompts are shown in Appendix X. These structural cues are similar to those in LLM pretraining data (e.g., column names in SQL tables), suggesting that pretraining on such data enables the model to recognize the structuring role of performance-critical tokens, allowing the representations to store higher-level information.

To assess the structuring characteristic of performance-critical tokens, we perturb the structure of one-shot ICL prompts in two stages, where the one-shot setting could eliminate repetition as a confounding factor. First, we disrupt the lexical meaning of template tokens, since token meaning helps LLMs distinguishing the different parts of a prompt. Then, we remove all template tokens to eliminate any source of structure cues. Table 8 shows that disrupting structural cues decreases performance, highlighting their importance. Consistent with Section 5.2, removing all template tokens results in 0% performance due to the complete elimination of structural cues. Supplemental experiments in Appendix W further support this from a representation-level perspective.

7 Conclusion

In this paper, we have provided a fine-grained characterization of performance-critical tokens, whose representations LLMs directly depend on to achieve high-level performance in ICL. Through a series of experiments, we have examined the roles of template tokens and stopword tokens within ICL as potential performance-critical tokens. Our findings add nuance to previous claims made about ICL, for example, that tokens other than label words could also provide valuable information directly affecting the performance. Overall, our results demonstrate that model performance depends directly on the presence of these tokens and that their lexical meaning, their repetition throughout the ICL prompt, and their structural formatting of ICL demonstrations are likely to play a role in how effectively they allow an LLM to recover the critical information needed to perform a task.

Ethics Statement

This work focuses on analyzing the working mechanisms of large language models and, as such, does not present any increased risks of harm beyond the existing norms of natural language processing or computational linguistics research. The associated risks include using a model trained on vast amounts of text, which may inadvertently contain biases. Another concern is the potential misuse of the model for generating misleading or harmful content. However, such a scenario is unlikely in our work, as we concentrate on classification tasks with fixed outputs.

Acknowledgements

The authors thank all the reviewers for their suggestions and comments. This work is supported by National Natural Science Foundation of China (No. U21B2009). Jackie Chi Kit Cheung is supported by Canada CIFAR AI Chair program.

References

Akyürek, E.; Schuurmans, D.; Andreas, J.; Ma, T.; and Zhou, D. 2022. What learning algorithm is in-context learning? Investigations with linear models. *ICLR*.

Bai, Y.; Chen, F.; Wang, H.; Xiong, C.; and Mei, S. 2023. Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection. *arXiv preprint arXiv: 2306.04637*.

Bai, Y.; Zou, X.; Huang, H.; Chen, S.; Rondeau, M.-A.; Gao, Y.; and Cheung, J. C. K. 2024. CItruS: Chunked Instruction-aware State Eviction for Long Sequence Modeling. *arXiv preprint arXiv: 2406.12018*.

Bertsch, A.; Ivgi, M.; Alon, U.; Berant, J.; Gormley, M. R.; and Neubig, G. 2024. In-Context Learning with Long-Context Models: An In-Depth Exploration. *arXiv preprint arXiv: 2405.00200*.

Bhattamishra, S.; Patel, A.; Blunsom, P.; and Kanade, V. 2023. Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions. *arXiv preprint arXiv:2310.03016*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *NeurIPS*, 33: 1877–1901.

Chen, Y.; Zhao, C.; Yu, Z.; McKeown, K.; and He, H. 2024. Parallel structures in pre-training data yield in-context learning. *arXiv preprint arXiv:2402.12530*.

Costa-jussà, M. R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 177–190. Springer.

De Marneffe, M.-C.; Simons, M.; and Tonhauser, J. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, 107–124.

Dong, Y.; Jiang, J.; Zhu, Z.; and Ning, X. 2025. Understanding Task Vectors in In-Context Learning: Emergence, Functionality, and Limitations. *arXiv preprint arXiv:2506.09048*.

Guo, T.; Hu, W.; Mei, S.; Wang, H.; Xiong, C.; Savarese, S.; and Bai, Y. 2023. How Do Transformers Learn In-Context Beyond Simple Functions? A Case Study on Learning with Representations. arXiv:2310.10616.

Han, X.; Simig, D.; Mihaylov, T.; Tsvetkov, Y.; Celikyilmaz, A.; and Wang, T. 2023. Understanding In-Context Learning via Supportive Pretraining Data. In *Proc. of ACL 2023, Vol. 1: Long Papers*, 12660–12673.

Hao, Y.; Sun, Y.; Dong, L.; Han, Z.; Gu, Y.; and Wei, F. 2022. Structured Prompting: Scaling In-Context Learning to 1,000 Examples. *arXiv preprint arXiv: 2212.06713*.

Hendel, R.; Geva, M.; and Globerson, A. 2023. In-Context Learning Creates Task Vectors. arXiv:2310.15916.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Jiang, G.; JIANG, C.; Li, Z.; Xue, S.; ZHOU, J.; Song, L.; Lian, D.; and Wei, Y. 2025. Unlocking the Power of Function Vectors for Characterizing and Mitigating Catastrophic Forgetting in Continual Instruction Tuning. In *ICLR 2025*.

Lenartowicz, A.; Simpson, G. V.; Haber, C. M.; and Cohen, M. S. 2014. Neurophysiological signals of ignoring and attending are separable and related to performance during sustained intersensory attention. *Journal of cognitive neuroscience*, 26(9): 2055–2069.

Li, M.; Gong, S.; Feng, J.; Xu, Y.; Zhang, J.; Wu, Z.; and Kong, L. 2023a. In-Context Learning with Many Demonstration Examples. *arXiv preprint arXiv: 2302.04931*.

Li, T.; Zhang, G.; Do, Q. D.; Yue, X.; and Chen, W. 2024. Long-context LLMs Struggle with Long In-context Learning. *TMLR*.

Li, Y.; Ildiz, M. E.; Papailiopoulos, D.; and Oymak, S. 2023b. Transformers as Algorithms: Generalization and Stability in In-context Learning. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *Proc. of ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, 19565–19594. PMLR.

Lin, Z.; and Lee, K. 2024. Dual operating modes of in-context learning. In *ICML 2024*.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2021. What Makes Good In-Context Examples for GPT-3? *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

Liu, S.; Xing, L.; and Zou, J. 2023. In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. *arXiv preprint arXiv: 2311.06668*.

Liu, Z.; Desai, A.; Liao, F.; Wang, W.; Xie, V.; Xu, Z.; Kyrillidis, A.; and Shrivastava, A. 2023. Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache

Compression at Test Time. In Oh, A.; Neumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *NeurIPS 2023*, volume 36, 52342–52364. Curran Associates, Inc.

Loper, E.; and Bird, S. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In *Proc. of ACL 2022*, 8086–8098. Dublin, Ireland: Association for Computational Linguistics.

Madaan, A.; and Yazdanbakhsh, A. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Mao, H.; Liu, G.; Ma, Y.; Wang, R.; Johnson, K.; and Tang, J. 2024. A Data Generation Perspective to the Mechanism of In-Context Learning. *arXiv preprint arXiv: 2402.02212*.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? *arXiv preprint arXiv: 2202.12837*.

Pan, J.; Gao, T.; Chen, H.; and Chen, D. 2023. What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Findings of ACL 2023*, 8298–8319. Association for Computational Linguistics.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL 2002*, 311–318.

Rayner, K. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3): 372.

Sarica, S.; and Luo, J. 2021. Stopwords in technical language processing. *Plos one*, 16(8): e0254937.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proc. of NAACL-HLT 2019, Vol. 1 (Long and Short Papers)*, 4149–4158. Minneapolis, Minnesota: Association for Computational Linguistics.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Tikhonov, P.; Oseledets, I.; and Tutubalina, E. 2025. One Task Vector is not Enough: A Large-Scale Study for In-Context Learning. *arXiv preprint arXiv:2505.23911*.

Todd, E.; Li, M. L.; Sharma, A. S.; Mueller, A.; Wallace, B. C.; and Bau, D. 2023. Function Vectors in Large Language Models. arXiv:2310.15213.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Voorhees, E. M.; and Tice, D. M. 2000. Building a question answering test collection. In *Proc. of SIGIR 2000*, 200–207.

Wang, L.; Li, L.; Dai, D.; Chen, D.; Zhou, H.; Meng, F.; Zhou, J.; and Sun, X. 2023. Label Words are Anchors: An Information Flow Perspective for Understanding In-Context Learning. *Proc. of EMNLP 2023*.

Wilbur, W. J.; and Sirotkin, K. 1992. The automatic identification of stop words. *Journal of information science*, 18(1): 45–55.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2021. An Explanation of In-context Learning as Implicit Bayesian Inference. *International Conference on Learning Representations*.

Yan, J.; Xu, J.; Song, C.; Wu, C.; Li, Y.; and Zhang, Y. 2024. Understanding In-Context Learning from Repetitions. *ICLR*.

Yang, L.; Lin, Z.; Lee, K.; Papailiopoulos, D.; and Nowak, R. 2025. Task vectors in in-context learning: Emergence, formation, and benefit. *arXiv preprint arXiv:2501.09240*.

Yin, F.; Vig, J.; Laban, P.; Joty, S.; Xiong, C.; and Wu, C.-S. 2023. Did You Read the Instructions? Rethinking the Effectiveness of Task Definitions in Instruction Learning. In *Proc. of ACL 2023*, 3063–3079. Toronto, Canada: Association for Computational Linguistics.

Yoo, K. M.; Kim, J.; Kim, H. J.; Cho, H.; Jo, H.; Lee, S.-W.; goo Lee, S.; and Kim, T. 2022. Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations. *arXiv preprint arXiv: 2205.12685*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *NeurIPS*, volume 28. Curran Associates, Inc.

Zhang, Y.; Feng, S.; and Tan, C. 2022. Active Example Selection for In-Context Learning. In *Proc. of EMNLP 2022*, 9134–9148.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; Wang, Z.; and Chen, B. 2023. H_2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. *arXiv preprint arXiv: 2306.14048*.

Zhao, A.; Ye, F.; Fu, J.; and Shen, X. 2024. Unveiling In-Context Learning: A Coordinate System to Understand Its Working Mechanism. *arXiv preprint arXiv:2407.17011*.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In Meila, M.; and Zhang, T., eds., *ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, 12697–12706. PMLR.

Zhou, Y.; Li, J.; Xiang, Y.; Yan, H.; Gui, L.; and He, Y. 2023. The Mystery of In-Context Learning: A Comprehensive Survey on Interpretation and Analysis. *arXiv preprint arXiv: 2311.00237*.