Superfluous Instruction: Vulnerabilities Stemming from Task-Specific Superficial Expressions in Instruction Templates

Toma Suzuki, Yusuke Sakai, Justin Vasselli, Hidetaka Kamigaito, Taro Watanabe
Nara Institute of Science and Technology (NAIST), Japan
{suzuki.toma.ss5, sakai.yusuke.sr9, vasselli.justin_ray.vk4,
kamigaito.h, taro}@is.naist.jp

Abstract

Large language models (LLMs) achieve high performance through instruction-tuning, which involves learning various tasks using instruction templates. However, these templates often contain *task-specific expressions*, which are words that frequently appear in certain contexts but do not always convey the actual meaning of that context, even if they seem closely related to the target task. Biases inherent in such instruction templates may be learned by LLMs during training, potentially degrading performance when the models encounter superficial expressions. In this study, we propose a method that incorporates additional instructions to FLAN templates, without altering the base instruction to produce **"superfluous instructions"**. This allows us to investigate the vulnerabilities of LLMs caused by overfitting to task-specific expressions embedded in instruction templates. The experimental results revealed that the inclusion of superficial words strongly related to each task in the instruction text can alter the output, regardless of the intended meaning.

1 Introduction

Large language models (LLMs) adopt a training method called instruction-tuning (Wei et al., 2022a; Longpre et al., 2023), which enables them to respond appropriately to a wide range of user queries based on given instructions. To perform instruction-tuning, it is necessary to construct datasets consisting of instruction-output pairs. Instruction templates are typically designed to structure existing natural language processing tasks so that generative LLMs can produce relevant outputs. Furthermore, diverse templates for each task are crucial to avoid overfitting to any single template. Providing multiple templates during instruction-tuning is important for improving the model's generalization (Sakai et al., 2024). However, templates designed for specific tasks often contain task-specific words, which may introduce biases related to those tasks. Table 1 presents the five most significant words, based on TF-IDF (Ramos, 2003), for each task in the instruction template dataset FLAN (Wei et al., 2022a), showing a strong connection between the words used in the templates and their associated tasks.

	trivia qa	wmt16 translate	multi news	math dataset	true case
1	**answer**	**translate**	**article**	**problem**	**capitalize**
2	**question**	to	summary	**math**	**case**
3	the	language	this	**solution**	proper
4	**trivia**	in	true	**solve**	correctly
5	be	not	context	the	**low**

Table 1: The five most words with high TF-IDF scores in instruction templates for each task in the FLAN dataset.

In this study, we focus on surface-level biases arising from the presence of task-specific words in instruction templates. By leveraging FLAN, a widely adopted instruction template dataset that allows for precise control over word occurrences, we can rigorously evaluate the influence of such task-specific words. Furthermore, we propose **"superfluous instructions"** which incorporate unrelated text into FLAN templates, while preserving the original task-solving intent of the instructions. For example, we add expressions such as *"Answer the following question* ***without generating unrelated text****"*. These expressions are carefully designed not to interfere with the original intent. Therefore, we expect that they will not affect the model's output from a task-solving perspective.

We evaluated three models tuned by FLAN instructions using 80 superfluous instructions tailored to each task. The results show that adding superfluous instructions, particularly those containing task-specific superficial expressions, negatively impacted performance. This suggests that instruction-tuned LLMs are vulnerable to superficial cues in the instructions, which degrade performance even when the instruction's meaning remains unchanged.

Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM), pages 140–152

These findings provide important insights for developing more robust instruction-tuning methods.

2 Background and Related Work

Instruction-Tuning Datasets. FLAN (Wei et al., 2022a; Longpre et al., 2023) is a widely used English resource for instruction-tuning, designed to cover a broad range of natural language processing tasks. By adapting these templates to each task, diverse data can be generated for instruction tuning. In addition to FLAN, other datasets have been proposed that use different templates for instruction-tuning (Wang et al., 2022; Zhang et al., 2023; Chen et al., 2024). However, there are concerns that datasets created using templates might merely lead models to memorize the superficial patterns of the templates (Kung and Peng, 2023). As a result, LLMs may struggle to follow instructions that deviate from the patterns found in their training data, failing to produce the expected output. Alternatives to template-based approaches include generating instruction-tuning data from LLM outputs (Xu et al., 2024, 2023; Peng et al., 2023), or efficiently producing large datasets through methods like crowdsourcing (Wang et al., 2022; Mishra et al., 2022; Köpf et al., 2023). However, such data can inherit generation biases from the LLMs used (Kavumba et al., 2022; Zellers et al., 2019; Tamborrino et al., 2020; Omura et al., 2020) or include low-quality artifacts from crowdsourcing, known as Annotation Artifacts (Gururangan et al., 2020; Poliak et al., 2018; Tsuchiya, 2018). Training models with such data may cause them to develop strong biased responses toward certain characteristic words.

Vulnerabilities to Specific Instructions. LLMs can achieve enhanced performance through prompt engineering (Wei et al., 2022b; Kojima et al., 2022; Zhong et al., 2023; Yang et al., 2024; Zhou et al., 2023; Yao et al., 2023; Chen et al., 2025), or via prompt tuning (Lester et al., 2021; Liu et al., 2024; Li and Liang, 2021). While well-designed prompts can maximize their potential, there is also a concern that language models might not understand the meaning of the text but rather rely on characteristic tokens in the input, guiding their outputs solely based on the superficial expressions of prompts (Du et al., 2023; Kavumba et al., 2022; Zellers et al., 2019; Tamborrino et al., 2020; Omura et al., 2020; Zheng et al., 2025). This issue has also drawn attention from the perspective of instruction-following (Moon et al., 2025; Sakai et al., 2025; Qin et al., 2024; Zeng et al., 2024), consistency (Sakai et al., 2024; Lee et al., 2025; Raj et al., 2025), and safety (Dong et al., 2024; Li et al., 2024). Thus, while specific tokens can enhance a model's performance, they may also cause the model to behave differently than usual when encountering certain tokens. For instance, popular instruction-tuning datasets like FLAN include only positive instructions in their templates. As a result, it has been questioned whether language models can properly handle instructions involving negation, such as *"does not contain the keyword"* or *"does not imply the meaning"* (Kassner and Schütze, 2020; Jang et al., 2023; Hosseini et al., 2021; Hossain et al., 2020; Ye et al., 2023). These studies evaluated models' ability to reverse answers in tasks like NLI (Williams et al., 2018) by making minor changes to evaluation templates, e.g., replacing "plausible" with "implausible" or "correct" with "incorrect." Their findings suggest that language models struggle with handling negation. However, these analyses focus on introducing negation by simply replacing words in templates, which leaves it unclear whether LLMs are inherently vulnerable to semantic negation, or merely biased due to the disproportionate presence of positive over negative instructions in training templates.

3 Superfluous Instructions

We introduce "superfluous instructions" that contain target words for analysis but provide no new semantic information. By adding superficial expressions without semantic changes, we investigate how superficial expressions, such as task-specific words, affect model output. We use FLAN (Wei et al., 2022a)[1] as seed instruction templates.

3.1 Design of Base Superfluous Instructions

Superfluous instructions are phrases added to instructions in a way that does not change their meaning. For instance, the phrase *"without generating unrelated text"* is a superfluous instruction in Figure 1. Such phrases are natural yet do not alter the purpose of the tasks due to the presence of a double negative. To generalize this structure, we create variations such as *"without generating **{unrelated} {text}**"*, where ***{unrelated}*** is replaced with synonyms and ***{text}*** with task-specific words.

[1] https://github.com/google-research/FLAN/blob/main/flan/templates.py

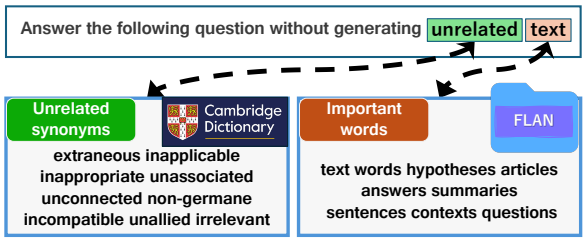

Figure 1: Base template of the superfluous instruction. The superfluous phrase **"without generating {unrelated} {text}"** includes placeholders, where **{unrelated}** is replaced with adjectives and **{text}** with nouns, using all possible combinations from the respective candidate sets. This allows us to add superficial expressions without introducing any semantic changes.

This approach allows us to generate multiple superfluous instructions per task. Since the core task instruction remains unchanged, the model's output should, in theory, also remain the same. If the output changes, it suggests that the superfluous instruction is influencing the model's behavior. For simplicity, "superfluous instructions" refers to the entire instructions containing the superfluous phrase: **"without generating *{unrelated} {text}*"**.

3.2 Word Selection for {Unrelated} Part

We fill the **{unrelated}** placeholder in the base superfluous instruction with synonyms of the word "unrelated" to evaluate the model's ability to generalize. By comparing the results across multiple instructions, we assess how the model responds to variations in the instruction. To identify appropriate synonyms, we consulted the Cambridge Dictionaries Online[2] and found 11 synonyms for "unrelated". We used 10 synonyms[3]: "unrelated," "extraneous," "inapplicable," "irrelevant," "unassociated," "incompatible," "unconnected," "unallied," "non-germane," and "inappropriate." We confirmed with native English speakers that all 10 variations are grammatically correct and preserve the original instruction's meaning. We then generated multiple instructions by replacing the ***{unrelated}*** placeholder in the phrase *"without generating **{unrelated}** text"* with each of these synonyms.

3.3 Important Word Selection from Instruction Templates

We replaced the **{text}** placeholder in the superfluous instruction with task-specific important words from each instruction to evaluate their effect on model performance. To identify these important words, we used TF-IDF (Ramos, 2003). For each task, we treated the set of templates associated with that task as a single document and computed TF-IDF scores. Since instruction tuning aims to improve model performance across multiple tasks, it is important to consider word importance not only within individual tasks but also across all templates. The TF-IDF calculation of our study is as follows:

$$tf(t,d) = \frac{n_{t,d}}{\Sigma_{s\in d} n_{s,d}}, \text{where } d \in D, \quad (1)$$

$$df(t,D) = |\{d \in D : t \in d\}|, \quad (2)$$

$$idf(t,D) = \log \frac{|D|}{df(t,D)} + 1, \quad (3)$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D). \quad (4)$$

Here, D denotes the collection of documents, d is a single document, t is the target word, and n is the raw count of the word t in d. For our TF-IDF calculation, we treat the entire collection of templates as D, where each task d_i is considered a document consisting of multiple templates. Each individual template within a task is denoted as d_{ij}.

Next, the TF-IDF scores for each word t were summed across the dataset D. To reduce bias from differences in word usage across tasks, we normalized these scores by dividing the sum by the number of tasks in which the word appears:

$$Importance(t,d_i,D) = \frac{\Sigma_{j=1}^{N} tfidf(t,d_{ij},D)}{df(t,D)}. \quad (5)$$

This approach balances *word importance* across the dataset while mitigating bias from infrequent words. We calculated TF-IDF scores after lemmatizing[4] the words in each template. The importance of each word, based on its TF-IDF score, is normalized by its occurrence count, as shown in Equation 5. However, words that appear very infrequently may yield artificially high importance scores. To address this, we consider only words with above-average occurrence counts. We define such frequently occurring words across the FLAN templates as high-importance words (henceforth, "important words").

Table 2 shows the top 15 important words. As indicated in Table 2, some of these words belong to parts of speech other than nouns. Therefore, to

[2] https://dictionary.cambridge.org/

[3] We exclude "foreign" because it did not strongly align with the meaning of "unrelated."

[4] For lemmatization, we used the "en_core_web_sm" model from the spaCy library: https://spacy.io/.

Rank	Word	TF-IDF	Importance
1	same	2.877	0.4795
2	**question**	7.601	0.4751
3	**hypothesis**	2.245	0.4491
4	**article**	5.226	0.4355
5	**answer**	6.246	0.3123
6	**summary**	2.426	0.2696
7	true	2.032	0.2540
8	we	1.800	0.2250
9	if	2.185	0.2185
10	two	1.706	0.2133
11	**word**	1.268	0.2114
12	next	2.240	0.2037
13	**sentence**	7.615	0.1953
14	**context**	1.533	0.1917
15	paragraph	1.519	0.1898

Table 2: Top 15 words that appear more frequently than average and have high importance scores. Words highlighted in bold were used in this study. Note that words with high TF-IDF scores do not always have high importance scores, e.g., "sentence".

maintain the correct structure of the superfluous instruction, we selected only nouns with an importance score of 0.19 or higher. The final eight words used in our experiments are highlighted in bold in Table 2. For consistency, countable nouns were used in their plural forms.

4 Experimental Setup

LLMs. We used three instruction-tuned LLMs based on FLAN templates, with different parameter sizes: FLAN-T5 XL (3B) based on T5-XL (Raffel et al., 2020); FLAN-T5 XXL (11B) based on T5-XXL (Raffel et al., 2020); FLAN-UL2 (20B) (Tay et al., 2023) based on UL2 (Chung et al., 2024).

Datasets. We selected MMLU (Hendrycks et al., 2021) and BBH (Suzgun et al., 2023). MMLU covers 57 subjects with varying difficulty, including STEM, law, medicine, and ethics. BBH focuses on 23 particularly challenging tasks for language models, derived from the broader BIG-Bench dataset (Srivastava et al., 2023), which spans 204 categories, including linguistics and software knowledge. These datasets are reserved for evaluation and not trained for each model.

Evaluations. We used 8-bit quantized inferences (Dettmers et al., 2022) with greedy decoding in a zero-shot setting[5]. We evaluated the models using accuracy as the evaluation metric. We apply simple post-processing to remove whitespace and newline characters, convert the text to lowercase, and then evaluate using exact match accuracy.

[5]This was implemented using HuggingFace Transformers (Wolf et al., 2020) and used a single A6000 GPU.

5 Experimental Results

Table 3 shows the evaluation scores for each model and task with superfluous instruction.

5.1 Effect of Adding Superfluous Instructions

In Table 3, where the **{unrelated}** part of the instructions was replaced with synonyms, all models exhibited a performance drop compared to the standard instructions, indicating that superfluous instructions negatively impact performance. BBH showed a larger score decrease than MMLU, which can be attributed to BBH's more varied answer formats. This suggests that the models are highly fitted to the concise style of FLAN instructions and struggle to handle the redundancy introduced by the added phrases. Furthermore, contrary to expectations based on scaling laws, the standard deviation increased with larger model sizes. This suggests that improving generalization requires not only scaling up model size, but also careful selection of instruction templates.

5.2 Impact of Superfluous Instructions with Important Words

In Table 3, when the **{text}** part was replaced with important words, the scores dropped even further. This suggests that the presence of important words in FLAN templates can introduce vulnerabilities, affecting model behavior regardless of context. As in Section 5.1, the score drop was larger for BBH than for MMLU and became more pronounced with increasing model size. These results further support the hypothesis of overfitting to instruction templates, as discussed in Section 5.1.

5.3 Impact of Combining Superfluous Instructions and Important Words

When both **{unrelated}** and **{text}** were replaced, the score drops, with FLAN-T5 XXL and FLAN-UL2 being as high as when only the {text} part was replaced. This suggests that replacing important words {text} consistently led to substantial performance degradation, regardless of the accompanying {unrelated} term. Additionally, although BBH features tasks with diverse answer formats, while MMLU consists solely of multiple-choice questions, MMLU exhibited higher standard deviations. This indicates that replacing important

Replacement {unrelated}	Replacement {text}	Score	FLAN-T5 XL MMLU	FLAN-T5 XL BBH	FLAN-T5 XXL MMLU	FLAN-T5 XXL BBH	FLAN-UL2 MMLU	FLAN-UL2 BBH
Standard Instruction		acc.	47.1	33.7	52.5	41.0	53.1	34.5
✓		acc.	46.8±0.3	30.3±3.0	49.1±2.4	33.1±3.6	48.8±5.1	20.9±5.6
✓		Δ	↓ 0.3 (0.7%)	↓ 3.4 (10.1%)	↓ 3.4 (6.5%)	↓ 8.0 (19.4%)	↓ 4.4 (8.2%)	↓ 13.5 (39.3%)
	✓	acc.	45.8±1.5	26.3±4.3	45.8±9.1	31.2±5.0	33.3±14.3	14.7±8.9
	✓	Δ	↓ 1.3 (2.7%)	↓ 7.4 (22.1%)	↓ 6.7 (12.8%)	↓ 9.9 (24.1%)	↓ 19.8 (37.3%)	↓ 19.7 (57.2%)
✓	✓	acc.	46.3±1.5	27.3±4.8	46.9±6.3	30.7±5.7	37.1±13.8	16.7±8.8
✓	✓	Δ	↓ 0.8 (1.7%)	↓ 6.4 (19.0%)	↓ 5.6 (10.7%)	↓ 10.4 (25.3%)	↓ 16.1 (30.2%)	↓ 17.8 (51.6%)

Table 3: Average scores per model and instruction type across tasks. Checkmarks indicate which part of the instruction *"Answer the following question without generating **{unrelated} {text}**."* was replaced. When present, a checkmark means {unrelated} was replaced with synonyms and {text} with important words. The ± symbol denotes the standard deviation, and Δ indicates the change in score relative to the version with no replacements.

Replacement: {text}	MMLU	BBH
Standard Instruction	53.1	34.5
words	↓ 22.0 (41.4%)	↓ 14.7 (42.6%)
hypotheses	↓ 22.3 (41.9%)	↓ 21.3 (61.9%)
articles	↓ 16.3 (30.6%)	↓ 21.3 (61.7%)
answers	↓ 1.2 (2.2%)	↑ 0.6 (1.8%)
summaries	↓ 3.7 (7.0%)	↓ 17.8 (51.6%)
sentences	↓ 21.5 (40.4%)	↓ 21.9 (63.5%)
contexts	↓ 18.2 (34.2%)	↓ 23.5 (68.1%)
questions	↓ 35.1 (66.1%)	↓ 26.6 (77.1%)

Table 4: FLAN-UL2's average scores for each replaced important word across all **{unrelated}** replacements.

words disrupted the model's ability to select correct answers, even in the constrained format of multiple-choice tasks. These findings suggest potential overfitting to the instruction templates used during tuning. Moreover, contrary to expectations from scaling laws, FLAN-T5 XL showed smaller variations in score and standard deviation compared to FLAN-T5 XXL and FLAN-UL2, reinforcing the idea that improving generalization depends not only on model size, but also on factors such as the instruction templates used during tuning.

5.4 Analysis of the Relationship Between Important Words and Scores

To identify which important words had the greatest impact on performance, Table 4 presents FLAN-UL2's average scores for each replaced important word, averaged over all {unrelated} replacements. The word "answers" caused the smallest change in scores, suggesting minimal influence on model behavior. In contrast, "questions" led to the largest score drop in both BBH and MMLU. Additionally, while "summaries" had little effect on MMLU, it caused a noticeable drop in BBH, similar to the behavior observed when using "text" in the base instruction. In summary, compared to both the standard instruction and the basic "text" prompt, the use of important words resulted in larger score decreases, confirming that these words have a strong influence on model behavior.

6 Discussion

6.1 Analysis of Score Decrease by Each Task

To understand how each important word affects model behavior, we analyzed task-level score changes in MMLU and BBH. Figures 2 and 3 show the scores without the superfluous instruction (w/o), and with replacements to {unrelated} (U), {text} (T), or both (U/T). Tasks are ordered by the standard deviation of scores across these conditions, from highest (top left) to lowest (bottom right).

MMLU. In most MMLU tasks shown in Figure 2, the scores for all three models are quite similar when standard instructions (column w/o) are used. However, superfluous instructions lead to noticeable variations in scores across tasks. For tasks with high standard deviation (top left), FLAN-UL2 (green line) shows a significant score drop when the prompt is altered. Similarly, FLAN-T5 XXL also shows a decline, especially in tasks with greater score variability, while FLAN-T5 XL exhibits minimal score changes. We also examined the impact of replacing {unrelated} and {text}. For FLAN-UL2, scores declined when {unrelated} was replaced, but an even larger drop occurred when {text} was substituted. This suggests that, for certain tasks, replacing {text} has a greater impact on performance than replacing {unrelated}.

BBH. In BBH tasks shown in Figure 3, even with standard instructions (column w/o), score trends varied across models, in contrast to the MMLU

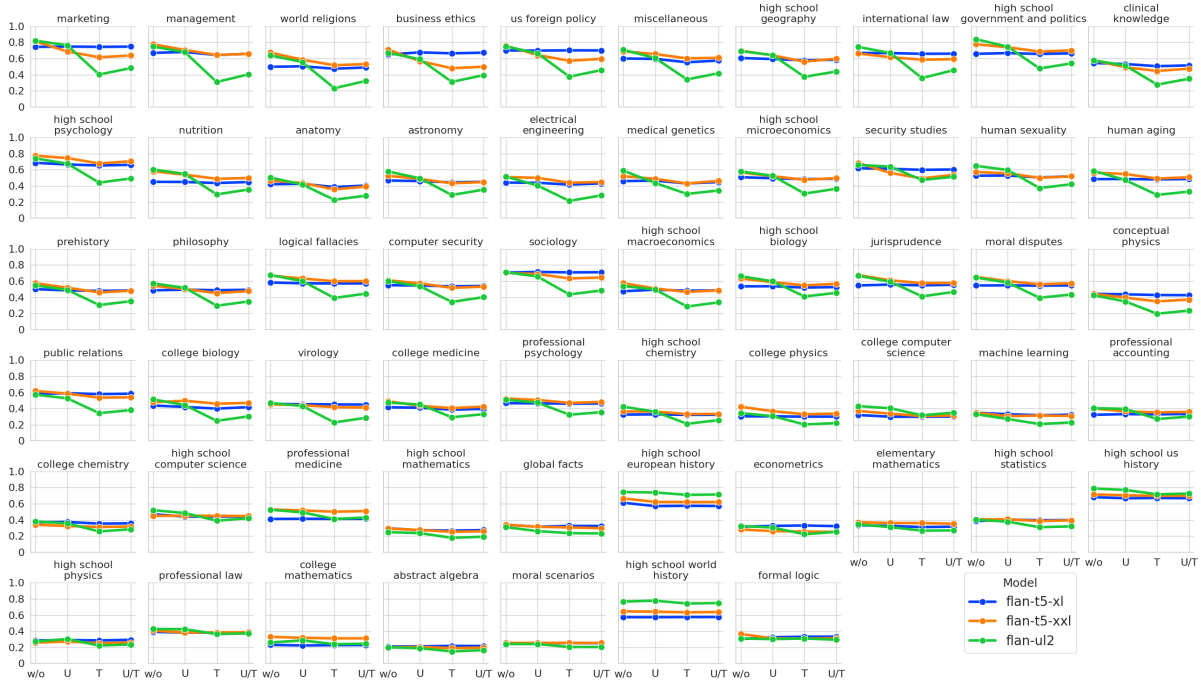

Figure 2: Accuracy for each task in MMLU. "w/o" indicates values without superfluous instructions, "U" indicates values with changes to {unrelated}, "T" indicates changes to {text}, and "U/T" indicates changes to both. Results are arranged from top left to bottom right in order of decreasing standard deviation for each task.

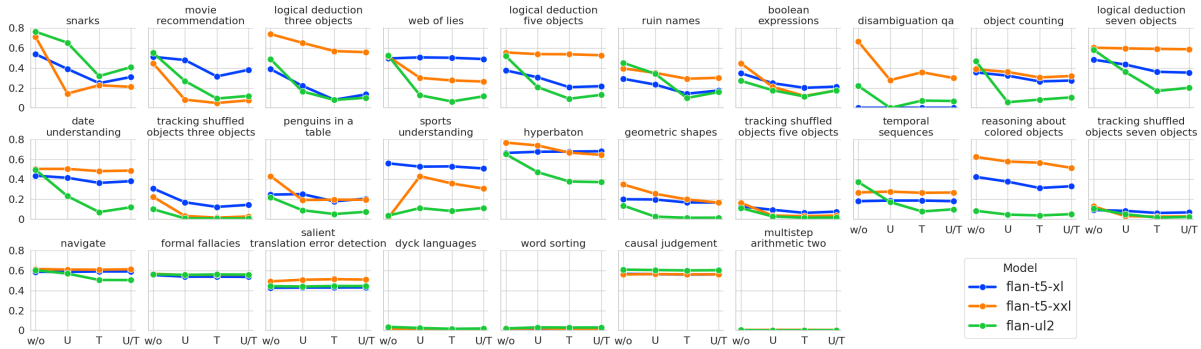

Figure 3: Accuracy for each task in BBH. "w/o" indicates values without superfluous instructions, "U" indicates values with changes to {unrelated}, "T" indicates changes to {text}, and "U/T" indicates changes to both. Results are arranged from top left to bottom right in order of decreasing standard deviation for each task.

case. Additionally, the score variations introduced by superfluous instructions were quite diverse. For FLAN-UL2, replacing {text} led to substantial score drops in many tasks. However, in tasks such as snarks and movie recommendation, the drop from {unrelated} replacements was relatively small compared to {text}, indicating that {text} played a stronger role in influencing model behavior in these tasks. For FLAN-T5 XXL, tasks such as snarks, disambiguation_qa, and sports understanding showed higher scores when {text} was replaced than when {unrelated} was, suggesting that important words had a positive effect on performance in these cases. FLAN-T5 XL, similar to its performance on MMLU, showed relatively little change in scores. The four tasks with the lowest standard deviations, except for causal judgement, consistently showed low accuracy across all prompts, indicating their difficulty for the models. BBH appears to contain many tasks that are highly sensitive to prompt variations. While MMLU consists entirely of multiple-choice questions (A to D), BBH includes tasks with various answer formats, such as valid-invalid, true-false, sorting, and symbol-based answers, leading to substantial variation in response quality depending on the prompt.

Case Analysis 1: To clarify how the replaced important words specifically impacted model behavior, we conducted a detailed analysis of several tasks from MMLU and BBH where the score decreased more significantly when {text} was replaced than when {unrelated} was replaced. We also examined BBH tasks that exhibited notable changes. For example, in the movie recommendation task, FLAN-UL2's score decreased further when important words were replaced after adding superfluous instructions. The frequent occurrence of the word "movie" in this task, which also appears in some FLAN template tasks, may suggest overfitting. While the movie recommendation task involves label selection, some FLAN template tasks require summary or sentence generation, often using words like "summarize" or "sentence". This overlap in terminology likely contributed to overfitting, resulting in a substantial drop in performance.

Case Analysis 2: Another noteworthy example is the sports understanding task. FLAN-T5 XL achieved around 60% accuracy, but the larger models, FLAN-T5 XXL and FLAN-UL2, showed lower performance even with standard instructions. Interestingly, FLAN-T5 XXL's score improved to 40% when superfluous instructions were added. In this task, the word "plausible" appears frequently, and the correct responses are "yes" or "no". The FLAN template task Copa also uses "plausible", but it involves multiple-choice answers. With standard instructions, FLAN-T5 XXL often responded in choice format, e.g., "(II)", but with superfluous instructions, correct yes-no responses increased. This suggests that FLAN-T5 XXL was overfitting to the word "plausible" in the prompt, and that the insertion of superfluous expression helped reduce this overfitting. These observations further support the hypothesis of word-level overfitting within the FLAN templates. This overfitting appears to influence both score performance degradation and improvement, depending on the specific task and prompt structure.

6.2 Impact of Low-Importance Words

Motivation and Settings. We examine whether the performance decrease attributed to high-importance words in Section 3.3 can also be observed with "low-importance words". We define low-importance words as those ranked among the lowest in importance scores. Table 5 lists the words with low importance. The final seven noun words used in our experiments are highlighted in bold. For consistency, countable nouns were replaced with their plural forms. To test this, we created similar instructions using low-importance words and calculated task scores for each model and instruction type. From the bottom 36 words in importance listed in Table 5, the nouns that appear in the FLAN templates include: "messages," "potentials," "terms," "definitions," "examples," "opinions," and "contents". These words were substituted into the {text} part of the instructions, while the {unrelated} part was also replaced with its synonyms, resulting in a total of 70 generated superfluous instructions.

	Word	TF-IDF	Importance		Word	TF-IDF	Importance
1	your	0.043	0.043	19	give	0.523	0.065
2	means	0.043	0.043	20	one	0.917	0.065
3	out	0.043	0.043	21	otherwise	0.131	0.066
4	resemble	0.043	0.043	22	tell	0.536	0.067
5	closely	0.043	0.043	23	so	0.068	0.068
6	try	0.050	0.050	24	second	0.277	0.069
7	else	0.050	0.050	25	first	0.277	0.069
8	impossible	0.050	0.050	26	return	0.209	0.070
9	**messages**	0.053	0.053	27	type	0.070	0.070
10	**potentials**	0.053	0.053	28	at	0.142	0.071
11	propose	0.053	0.053	29	embody	0.071	0.071
12	**term**	0.225	0.056	30	**example**	0.356	0.071
13	generate	1.186	0.059	31	perceive	0.072	0.072
14	follow	2.255	0.063	32	**opinion**	0.072	0.072
15	here	1.157	0.064	33	whether	0.146	0.073
16	another	0.065	0.065	34	above	1.404	0.074
17	**definition**	0.065	0.065	35	think	0.151	0.075
18	both	0.065	0.065	36	**contents**	0.303	0.076

Table 5: List of 36 low-importance words, ranked by importance score from lowest to highest. The seven bolded nouns were used in our experiments.

Relationship Between Low-Importance Words and Scores. Table 6 presents the task scores for each model and instruction type. When using instructions with low-importance words, particularly in BBH, the rate of score decline tended to increase with larger model sizes. However, this decline was smaller for FLAN-UL2 compared to the case with high-importance words. Similar trends were observed in the other models, though the changes were generally smaller. Furthermore, Table 7 shows the average scores for each low-importance word. Except for "terms" and "definitions", most words caused only minimal score changes across all models, indicating limited impact on performance. However, "terms" and "definitions" led to substantial drops in FLAN-T5 XXL and FLAN-UL2, despite being classified as low-importance. This may be due to "definitions" appearing only once in the original FLAN templates used for

Replacement {unrelated}	Replacement {text}	Score	FLAN-T5 XL MMLU	FLAN-T5 XL BBH	FLAN-T5 XXL MMLU	FLAN-T5 XXL BBH	FLAN-UL2 MMLU	FLAN-UL2 BBH
Standard Instruction		acc.	47.1	33.7	52.5	41.0	53.1	34.5
	✓	acc.	46.8±0.2	27.6±3.3	46.3±7.0	33.1±6.7	48.3±7.5	24.2±7.1
	✓	Δ	↓ 0.2 (0.5%)	↓ 6.1 (18.2%)	↓ 6.2 (11.9%)	↓ 8.0 (19.4%)	↓ 4.8 (9.1%)	↓10.3 (29.9%)
	✓	acc.	46.9±0.4	28.6±3.4	47.7±7.0	34.1±6.6	48.8±6.9	25.0±7.0
✓	✓	Δ	↓ 0.2 (0.5%)	↓ 5.2 (15.3%)	↓ 4.9 (9.2%)	↓ 6.9 (16.9%)	↓ 4.4 (8.2%)	↓ 9.4 (27.4%)

Table 6: Average scores per model and instruction type across tasks using lower importance words. Checkmarks indicate which part of the instruction *"Answer the following question without generating **{unrelated} {text}**."* was replaced. When present, a checkmark means {unrelated} was replaced with synonyms and {text} with important words. The ± symbol denotes the standard deviation, and Δ indicates the change in score relative to the version with no replacements.

Replacement: {text}	MMLU	BBH
Standard Instruction	53.1	34.5
messages	↓ 0.6(1.2%)	↓ 5.8(16.8%)
potentials	↓ 0.5(1.0%)	↓ 2.3(6.7%)
terms	↓ 14.8(27.9%)	↓ 15.0(43.5%)
definitions	↓ 9.3(17.5%)	↓ 19.4(56.3%)
examples	↓ 2.6(4.9%)	↓ 8.5(24.6%)
opinions	↓ 0.8(1.5%)	↓ 5.5(16.0%)
contents	↓ 1.8(3.4%)	↓ 9.7(28.0%)

Table 7: FLAN-UL2's average scores for each replaced low important word across all {unrelated} replacements.

TF-IDF computation, but being used frequently in the natinst_v2 task included in the updated FLAN-v2 templates[6]. At the task level, FLAN-UL2 again showed greater score variability, consistent with observations for high-importance words. In MMLU, scores remained stable across different low-importance words, whereas BBH showed slightly more variation, though still less than when high-importance words were used. These results support the use of our importance score as an indicator of words that may cause overfitting.

6.3 Low-Importance Words by Tasks

Figures 4 and 5 show task-level results using low-importance words. Since the word "text" is not among the selected low-importance words, the "U" column contains no values. Tasks are ordered by standard deviation from top left to bottom right, following the same order as in Figures 2 and 3.

MMLU. In Figure 4, MMLU shows minimal score variation when low-importance words are used. When {text} is replaced (T), tasks that previously showed large drops with high-importance words now exhibit only slight decreases. When both {unrelated} and {text} are replaced (U/T), scores remain nearly the same as when only {text} is replaced, suggesting that {unrelated} has a limited impact. This trend aligns with the earlier results using high-importance words.

BBH. In Figure 5, similar patterns are observed. For most tasks, excluding "sports understanding", FLAN-T5 XL and FLAN-T5 XXL show little to no score change, in contrast to the greater variations seen with high-importance words in Figure 3. FLAN-UL2 displays some variability, but again, to a lesser extent. These results support the claim that high-importance words more strongly affect model behavior and task performance. Interestingly, in the "sports understanding" task, replacing {text} led to a score increase to about 30% for FLAN-T5 XXL, while FLAN-UL2 remained mostly unchanged. This contrasts with the high-importance condition in Figure 3, where FLAN-T5 XXL improved by about 40% and FLAN-UL2 by 10%. These findings highlight the importance of task-specific prompt design.

6.4 What Does Importance Score Capture?

We analyze why certain words that strongly influence model behavior tend to have high importance scores. First, we calculated TF-IDF scores within FLAN templates to assess how distinctive a word is in contexts requiring specific answer formats (Table 1). Next, we identified task-specific important words using the importance score and confirmed which words were generally characteristic across tasks (Figure 1). Finally, we filtered out words with high scores that appeared only a few times, as described in Section 3.3.

This process allowed us to **efficiently identify** words that are both strongly tied to output formats and frequently encountered during training. These

[6] https://github.com/google-research/FLAN/blob/main/flan/v2/templates.py

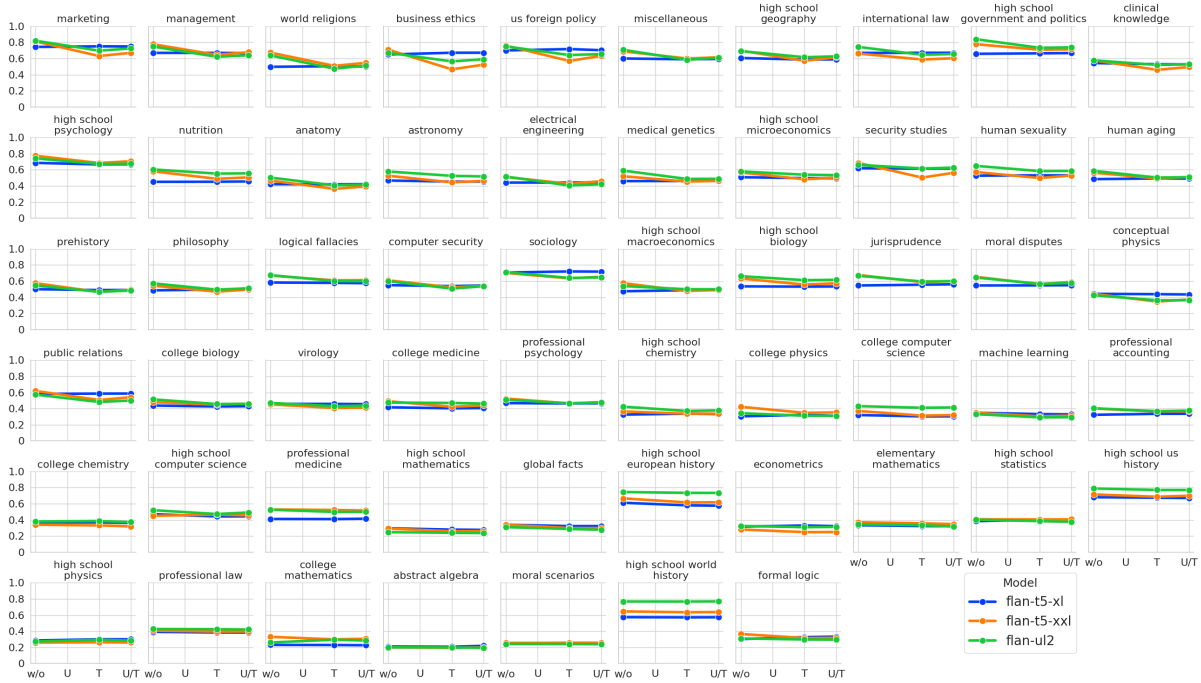

Figure 4: Accuracy for each task in MMLU. The notation and order in each table are the same as in Figure 2. "w/o" indicates values without superfluous instructions, "U" indicates values with changes to {unrelated}, "T" indicates changes to {text}, and "U/T" indicates changes to both. In contrast to Figure 2, the word "text" is not among the low TF-IDF words, so there are no values in the "U" column.

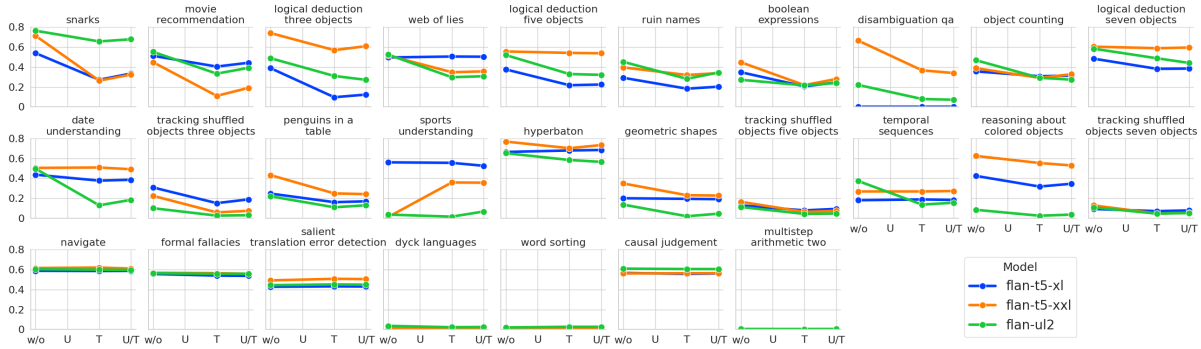

Figure 5: Accuracy for each task in BBH. The notation and order in each table are the same as in Figure 3. "w/o" indicates values without superfluous instructions, "U" indicates values with changes to {unrelated}, "T" indicates changes to {text}, and "U/T" indicates changes to both. In contrast to Figure 3, the word "text" is not among the low TF-IDF words, so there are no values in the "U" column.

results suggest that such words are more likely to cause overfitting and that heuristic methods like TF-IDF-based importance scores may be useful for identifying vulnerabilities in deep learning models.

7 Conclusion

In this study, we proposed a novel method for designing instruction templates to analyze the impact of task-specific superficial expressions found in instruction-tuning templates on the performance of large language models. Using this method, we generated instructions based on the FLAN templates and conducted evaluations on both MMLU and BBH tasks. The results revealed that the performance of LLMs is affected by task-specific superficial expressions included in the instructions. This insight is essential for developing more robust instruction-tuning methods. In future work, we plan to explore solutions such as replacing these superficial expressions during instruction-tuning to address the issues identified in this study.

8 Limitations

Language Models. Our study validated the findings using a limited set of open models instruction-tuned on the FLAN dataset. Due to resource constraints, we could not train models on the full FLAN datasets and instead relied on widely used instruction-tuned models. This choice allowed us to isolate the impact of word biases in the FLAN templates. However, our conclusions may not generalize to all large language models. Future work could involve comparisons among models with similar architectures to further examine these effects.

Generality of Dataset. In this study, we used only two datasets, MMLU and BBH, which were explicitly labeled as held-out tasks in the original paper (Longpre et al., 2023). This choice was made to create an ideal environment for isolating the influence of words in the FLAN templates by mitigating other variables. Whether the results observed with these two tasks can be replicated in other datasets remains an open question for future research. However, the in-depth analysis at the task level within BBH could help clarify the potential impact of similar effects in other datasets. In the future, we could also explore broader impacts in datasets like MMMLU[7], which includes multilingual tasks and might provide insights similar to those seen in our held-out tasks.

Datasets for Instruction-Tuning. Although we investigated the influence of word bias in templates, other methods have been developed to reduce word-related biases, such as allowing language models to generate diverse prompts (Wang et al., 2023; Taori et al., 2023; Kojima et al., 2021; Nayak et al., 2024). This approach may increase the variety of tasks and phrases used. However, as previous research has repeatedly shown, biases in the words and ideas produced by language models remain a concern. Techniques like TF-IDF, which count word frequency, continue to be effective in detecting such biases early on. Additionally, there are restrictions on how data generated by models like Llama2 (Touvron et al., 2023) can be used, such as limitations on usage outside of Llama 2 or derivative works[8]. Considering these constraints, instruction-tuning with templates remains valuable, and efforts to mitigate bias in this context are still essential.

[7] https://huggingface.co/datasets/openai/MMMLU

[8] https://github.com/metallama/llama/blob/main/LICENSE

9 Ethical Considerations

Dataset. We used public datasets and modified them. These datasets are allowed to be used and modified under their respective licenses. Therefore, our dataset does not raise ethical considerations.

Use of AI Assistants. In this study, we have used GitHub Copilot and ChatGPT as an AI assistant for coding and writing support.

References

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2025. Unleashing the potential of prompt engineering for large language models. *Patterns*, 6(6):101260.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024. InstructZero: Efficient instruction optimization for black-box large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6503–6518. PMLR.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. Attacks, defenses and evaluations for LLM conversation safety: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6734–6747, Mexico City, Mexico. Association for Computational Linguistics.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67(1):110–120.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312, Online. Association for Computational Linguistics.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Proceedings of The 1st Transfer Learning for Natural Language Processing Workshop*, volume 203 of *Proceedings of Machine Learning Research*, pages 52–62. PMLR.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2333–2352, Dublin, Ireland. Association for Computational Linguistics.

Noriyuki Kojima, Alane Suhr, and Yoav Artzi. 2021. Continual learning for grounded instruction generation by observing human following behavior. *Transactions of the Association for Computational Linguistics*, 9:1303–1319.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Alexandrovich Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Julian Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1317–1328, Toronto, Canada. Association for Computational Linguistics.

Noah Lee, Jiwoo Hong, and James Thorne. 2025. Evaluating the consistency of LLM evaluators. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10650–10659, Abu Dhabi, UAE. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. 2024. Evaluating the instruction-following robustness of large language models to prompt injection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 557–568, Miami, Florida, USA. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. Gpt understands, too. *AI Open*, 5:208–215.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Hyeonseok Moon, Jaehyung Seo, Seungyoon Lee, Chanjun Park, and Heuiseok Lim. 2025. Find the intention of instruction: Comprehensive evaluation

of instruction understanding for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5944–5964, Albuquerque, New Mexico. Association for Computational Linguistics.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12585–12611, Bangkok, Thailand. Association for Computational Linguistics.

Kazumasa Omura, Daisuke Kawahara, and Sadao Kurohashi. 2020. A method for building a commonsense inference dataset based on basic events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2450–2460, Online. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2025. Improving consistency in large language models through chain of guidance. *Transactions on Machine Learning Research*.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2025. Revisiting compositional generalization capability of large language models considering instruction following ability. *Preprint*, arXiv:2506.15629.

Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. 2024. Toward the evaluation of large language models considering score variance across instruction templates. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 499–529, Miami, Florida, US. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3878–3887, Online. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

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, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language

models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, and 16 others. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Mengyu Ye, Tatsuki Kuribayashi, Jun Suzuki, Goro Kobayashi, and Hiroaki Funayama. 2023. Assessing step-by-step reasoning against lexical negation: A case study on syllogism. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14753–14773, Singapore. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. 2025. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *The Thirteenth International Conference on Learning Representations*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *Preprint*, arXiv:2302.10198.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.