# Search Augmented Instruction Learning

**Hongyin Luo**[1*] **Tianhua Zhang**[2*] **Yung-Sung Chuang**[1] **Yuan Gong**[1]
**Yoon Kim**[1] **Xixin Wu**[2] **Helen Meng**[2] **James Glass**[1]
[1] MIT Computer Science and Artificial Intelligence Lab, Cambridge MA, USA
[2] CUHK Centre for Perceptual and Interactive Intelligence, Hong Kong SAR, China
hyluo@mit.edu, thzhang@link.cuhk.edu.hk

## Abstract

It is widely believed that connecting large language models with search engines can improve their transparency, truthfulness, and accessing to up-to-date information. However, we show that search grounding introduces new challenges to language models because of distracting, misleading, and untrustworthy information. To deal with these difficulties, we propose search-augmented instruction learning (SAIL), which allows a fine-tuned language model to source, denoise, and reason based on a mixed set of helpful and distracting search results. With an instruction tuning corpus, we collect search results for each training case from different search APIs and domains, and construct a new search-grounded training set containing *(instruction, grounding information, response)* triplets. We then fine-tune the `LLaMA-7B` model on the constructed training set. Since the collected search results contain distracting and disputing languages, the model needs to learn to ground on trustworthy search results, filter out distracting passages, and generate the target response. The search result-denoising process entails explicit trustworthy information selection and multi-hop reasoning, since the retrieved passages might be informative but not contain the instruction-following answer. Experiments show that the fine-tuned `SAIL-7B` model has a strong instruction-following ability, and it performs significantly better on transparency-sensitive tasks, including question answering and fact checking.

## 1 Introduction

Large language models (LLMs) have demonstrated many impressive capabilities, including zero-shot inference and few-shot in-context learning (Wei et al., 2022a). Recent research has shown that LLMs benefit from instruction tuning (Ouyang et al., 2022), and that such instruction-tuned LLMs significantly outperform plain LLMs on zero-shot language tasks (Peng et al., 2023). Instruction-tuned LLMs have shown an ability to generate both natural and programming languages following natural language guidance and requests. To achieve the same goal, a pretrained LLM needs a number of annotated examples as in-context learning prompts.

Despite their impressive behavior, LLMs have a number of issues, including obsolescence and non-transparency. Understandably, LLMs are trained with corpora constructed up to a certain time point. With this fixed, pretrained or fine-tuned model, subsequently occurring information cannot appear in any informed generation by the LLM. One way to update the knowledge in LLMs is to re-train the entire model with an updated training corpus. However, this would be costly and time-consuming.

In terms of transparency, the predictions of LLMs are opaque because generations are not grounded on trustworthy sources. It is possible for an LLM to generate undesirable language that looks like human-generated text, including misinformation, stereotypes, and toxic language (Zhang et al., 2023; Hartvigsen et al., 2022). Without providing legitimate sources for LLM generated texts, it is difficult to catch and avoid these undesirable LLM behaviors.

To overcome these difficulties, a straightforward solution is to connect LLMs to information retrieval systems, especially commercial search engines. By doing so, the LLM can ground its predictions on information retrieved from an up-to-date knowledge base, and the sources of the generations would be transparent to users. Before LLMs became large enough to memorize a significant amount of world knowledge, retrieval-based grounding had been heavily studied for open-domain question answering (Chen et al., 2017; Kwiatkowski et al., 2019; Guu et al., 2020). Recent LLMs have also shown the potential of using information retrieval tools, e.g., Toolformer (Schick

[*] Equal contribution. Code and processed data are available at https://github.com/luohongyin/SAIL.

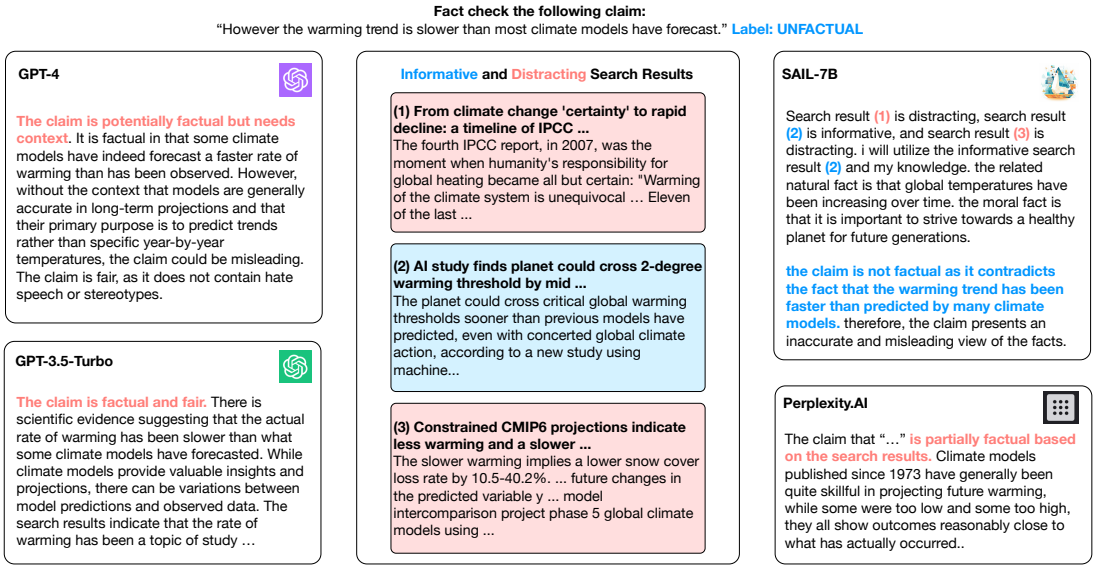


Figure 1: Fact checking grounding on complicated search results with SAIL-7B and strong commercial language models. The first and third passages are distracting since they do not contain information that supports or refutes the claim. And the first result on a 2007 report seems to be obsolete as answering climate-related questions usually needs up-to-date information. In contrast, the second passage disagrees with the claim by the statement "sooner than previous models have predicted". Although we acknowledge the complexity introduced by the term "most" in the claim, we contend that the second passage as most suitable. SAIL-7b successfully makes the the correct prediction while other commercial LLMs are distracted.

et al., 2023) and the ChatGPT (OpenAI, 2022) retrieval plugin. However, there remains a challenge: is there a trustworthy retrieval model and knowledge base that can be utilized by LLMs?

Existing studies on open-domain question answering have chosen Wikipedia as the *de facto* knowledge base that contains the answer to most questions. However, Zhang et al. (2023) found that the knowledge contained in Wikipedia is not sufficiently up-to-date nor complete for many tasks that require the latest knowledge, so grounding on Wikipedia might lead to worse answers than fully relying on LLMs. Another option is to leverage internet search engines, for example, Google, Bing, and DuckDuckGo.com[1].

Although widely used commercial search engines can index and retrieve a vast range of up-to-date information, their retrieval accuracy is ultimately limited, and third-party users cannot control the performance at the model level. As a result, retrieval results can be noisy, and unrelated information might be shown to users. This behavior suggests that there is a trade-off between deploying in-house retrieval systems and external search engines. Although it is possible to prompt LLMs to directly use the retrieval results, distracting search results can mislead the model and negatively influence the model's performance. As shown in Figure 1, ChatGPT is confused by a distracting passage and generates an incorrect fact check.

The challenges mentioned above are contradictory, and both have a negative impact on grounded language modeling with current LLMs - static knowledge bases and in-house retrievers are not sufficient or up-to-date for all tasks, while commercial search engines often generate distracting results. To address these challenges simultaneously, we propose a search-augmented instruction learning (SAIL) model. Given input instructions and contexts, the model is trained to generate high-quality responses according to the instruction grounding on the noisy research results. In other words, the model learns to denoise the retrieval results to generate high-quality responses.

In summary, we make the following contributions in this work:

1. We show that instruction-tuned LLMs can be heavily misled by distracting grounding information and noisy search results.
2. We constructed a search-augmented instruction training corpus.
3. We fine-tune a 7B-parameter language model (SAIL-7B) with the constructed training set,

[1]A free, privacy-preserving, zero-tracking search engine.

which outperforms strong baseline models including GPT-3.5-Turbo and Vicuna-13B on several NLP tasks.

By comparing the SAIL-7B model with LLaMA-7B, Vicuna-7B, GPT-3.5-turbo, and Vicuna-13B models on instruction following, question answering, and language checking tasks, we find that the SAIL-7B model has a strong instruction following ability and is robust against distracting grounding search results generated by different retrieval models. In addition, the SAIL model also achieves comparable performance to state-of-the-art instruction-following LLMs.

## 2 Method

### 2.1 Search Result Collection

In this work, we use the 52k self-instruction corpus created by the Alpaca team (Taori et al., 2023), and the corresponding responses generated by GPT-4 (Peng et al., 2023). For each instruction, we construct a search query by simply concatenating the instruction and the input, if any, and truncating the query to at most 60 words to fulfill the limitation of the search engine.

The constructed queries are fed into the DuckDuckGo search engine and the BM25 Wikipedia retriever, and the top three search results are retained. Each result consists of three fields: the title, a short piece of preview text, and the corresponding URL of the webpage. For simplicity, we do not further scrape the retrieved webpage, but just use the title and preview texts for further processing.

Each training example is assigned a list of corresponding search results. We pool the top-three DuckDuckGO and top-two BM25 search passages with Pyserini (Lin et al., 2021), a total of five search results. Among this pool, we randomly sample zero, one, two, and three search results with 20%, 20%, 20%, and 40% probability. Given this randomness, some training cases could be associated with search results from a single source.

### 2.2 In-context Retrieval Selection

To encourage the LLM to focus on trustworthy and informative search results, we concatenate a search filtering sequence before each annotated response of training instances. For example, *"Search result (1) is informative and search result (2) is distracting, so I will use the information from the search result (1)."*

However, the trustworthiness of each search result is not labeled, and the number of retrieval items is large. To bypass the need of costly human annotation, we employ an entailment classification model proposed in (Luo and Glass, 2023), which has in general been proven to be an effective approach for zero-shot setting (Obamuyide and Vlachos, 2018; Condoravdi et al., 2003; Ge et al., 2023). We feed each retrieved passage and the corresponding response into the entailment model and compare the entailed and contradictory scores. While most predictions are neutral against the response, the relation between entailed and contradictory scores can roughly indicate if a retrieved passage can provide useful information to generate the target response. As a result, we obtain pseudo-label *"search result (i) is informative"* if the entailed score is higher than the contradiction score, otherwise the search item is distracting. Note that our primary objective is not the construction of a human-labeled dataset comprising informative and distracting documents. Instead, we aim at proposing a label-free denoising method to enhance retrieval-augmented large language models.

Unlike the training stage, the absence of prior access to the target responses hinders the entailment model to predict pseudo-labels during inference. On the contrary, SAIL-7b acquires the capability to assess the relevance of various search results after training on both search results with pseudo-labels and the target responses. Subsequently, it can anchor the final response generation on informative search-augmentation. In other words, SAIL-7b would generate the search selection sequences (e.g., *"search result (i) is informative / distracting"*) before the final responses as shown in Figure 1.

### 2.3 Fine-tuning

After collecting the search results and generating in-context retrieval selection sequences, we construct input prompts following Figure 2 (b) with GPT-4 generated responses (Peng et al., 2023). Note that the most relevant retrieval result is located at the closest position to the instruction for the model to better use its information. We fine-tune LLaMA-7b models with the constructed prompts to generate both in-context retrieval selection sequences and annotated responses.

In practice, the models are fine-tuned with academic devices. Specifically, we use 4 $\times$ NVIDIA RTX A6000 GPUs (48GB $\times$ 4) to train the models

**(a) Standard Prompt**

```
Below is an instruction that describes a task.
Write a response that appropriately completes the request.

### Instruction: [Instruction]
### Input: [Input or None]

### Response:
```

**(b) Search-Augmented Prompt**

```
Below is an instruction that describes a task.
Write a response that appropriately completes the request.

### Related Information:
[Title 3]\n [Preview 3]
[Title 2]\n [Preview 2]
[Title 1]\n [Preview 1]

### Instruction: [Instruction]
### Input: [Input or None]

### Response:
```

Figure 2: Different prompting strategies used in this work. (a) **Standard prompt**: the prompt template used in Peng et al. (2023) to generate GPT-4 responses to the 52k instructions. (b) **Search-augmented prompt**: combining the top three search results and the instruction.

for 3 epochs. We apply mixed-precision training (fp16) with the standard AdamW optimizer. We set the maximum sequence length as 1,600 and the batch size as 32. Following Vicuna, we apply gradient checkpointing to reduce the memory cost. The entire fine-tuning process takes 24 hours ($24 \times 4$ GPU hours). To enable the fine-tuning, we applied gradient offload with Deepspeed and full-sharded data parallel (FSDP) (Paszke et al., 2019).

### 2.4 Evaluation

**SAIL for instruction following.** Following Peng et al. (2023), we evaluate the instruction following quality of different models by comparing with GPT-4 responses on the same set of instructions and scoring with GPT-4.

For each case, we construct an evaluation prompt by concatenating the instruction, the GPT-4 response, and the response of the target model. We feed the evaluation prompt to GPT-4 and ask it to score the two responses between 0 to 10. We use the Question-80[2] corpus (Chiang et al., 2023), which contains 80 questions to evaluate all models and we calculate the total score a model receives on all questions. To test the ability of the models on latest, unseen texts, we build another 80-question evaluation set based on latest news articles published after May 2023, which are never included in any training corpus as we finished all experiments. We name the new question set as New-Questions-80[3]. Because related information and knowledge are not included in the pretraining corpora, a language model has zero knowledge to answer and has to be grounded on an up-to-date search engine to generate informed answers.

We use the evaluation prompt authored by the Vicuna team[4]. The highest possible score is $80 \times 10 = 800$. It is worth noting that GPT-4 responses can receive slightly different scores against different counterparts. To normalize the difference, we calculate the ratio of model score / GPT-4 score for each test case as the final assessment as implemented in Peng et al. (2023).

**SAIL for Question Answering.** Besides evaluating the quality of instruction-guided generations, we assess the model's ability to answer commonsense questions. We also test the models on two different settings, including instructed zero-shot prediction and the search-augmentation mode. We evaluate the model performance on CommonsenseQA (CSQA; Talmor et al. (2019)), OpenbookQA (OBQA; Mihaylov et al. (2018)), and ARC-Challenge (Clark et al., 2018) benchmarks. All tasks require answering open-ended questions by selecting from a given set of candidate answers. Through the question-answering experiments, we show that instruction-tuned language models can be significantly biased by noisy research results.

**SAIL for Fact and Fairness Checking.** With the recent advances in LLMs that generate human-like languages without guaranteed alignment, human and machine-generated misinformation, stereotypes, and toxicity have become timely and significant concerns. Recent studies have shown that with appropriate instructions and prompts, LLMs can perform unified fact and fairness checking (Zhang et al., 2023). However, other attempts have relied only on LLMs, without grounding on any external sources, thus reducing the trustworthiness and transparency of the checking results.

In this work, we evaluate instructed fact and fairness checking, with the UniLC benchmark proposed in (Zhang et al., 2023), including Climate-Fever, PubHealth, Hate Speech Detec-

[2] https://github.com/lm-sys/FastChat/blob/main/fastchat/eval/table/question.jsonl

[3] Will release with code if accepted.

[4] https://github.com/lm-sys/FastChat/blob/main/fastchat/eval/table/prompt.jsonl

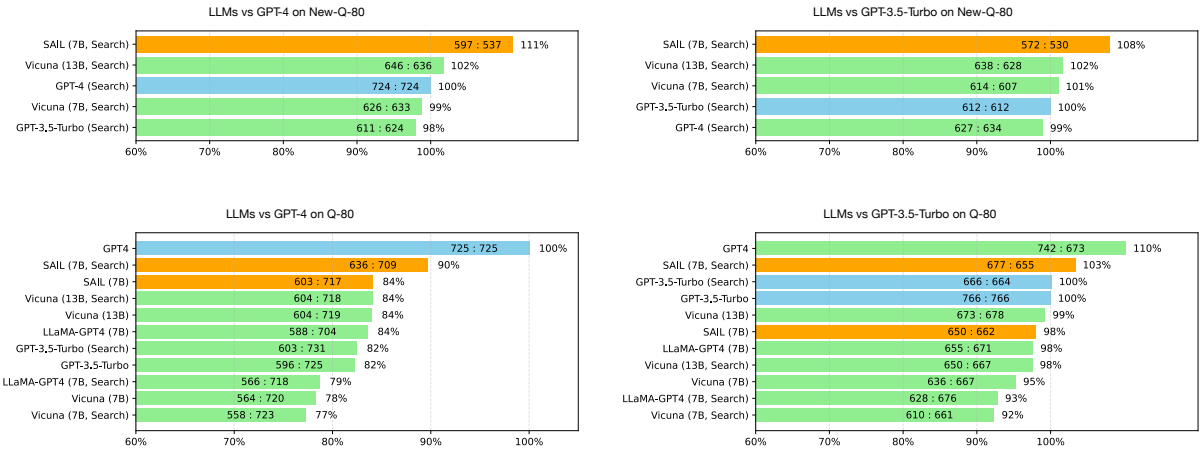


Figure 3: Scoring results of all language models on `Question-80` and `New-Question-80` benchmarks against `GPT-4` and `GPT-3.5-Turbo`. **Search** indicates generating responses with language models grounding on search results retrieved by `DuckDuckGO`, and **SAIL (7B)** stands for generating responses without search results, although the model is trained for grounded generations. Both `Vicuna-7 & 13B` are version 1.1 models. The orange bars stand for `SAIL-7B` and blue bars stand for `GPT-4` and `GPT-3.5-Turbo`.

tion, and Social Biase Frame (SBIC) tasks with two different settings - zero-shot and search-augmented. While we are not aware of what corpora are used to train GPT-4 and ChatGPT, we assess the language-checking performance of `Vicuna-7B-v1.1`, `Vicuna-13B-v1.1`, and `SAIL-7B` with and without search results.

## 3 Experiments

### 3.1 Instruction Following

**Automatic Evaluation with GPT-4.** We compare the performance of different models under end-to-end and search grounded settings against GPT-4 and ChatGPT models on `Question-80` and `New-Question-80`. The scoring results are shown in Figure 3.

By comparing to `GPT-4` on `Question-80`, we find that the search-augmented `SAIL-7B` model significantly outperforms all other models (90% vs <85%) using fewer training instructions and parameters, including strong baselines like `Vicuna-13B` and `GPT-3.5-Turbo` powered ChatGPT. This indicates that when the grounding information is provided, the model does not need as many parameters to memorize knowledge. In addition, the `SAIL-7B` model also achieves high performance even without search results, showing that the model performance is stable under different generation settings. Similar conclusions can be found by comparing all models against `GPT-3.5-Turbo`. While `GPT-4` is still better, experiment results show that the search-augmented `SAIL-7B` model achieves 103% of `GPT-3.5-Turbo` performance and the no-augmentation SAIL model achieves 98%, outperforming several strong baselines, including LLaMA tuned on GPT-4 instructions and Vicuna models with the same number of parameters. Besides `GPT-4`, search-augmented `SAIL-7B` is the only model that outperforms `GPT-3.5-Turbo` on both experiments.

In addition, we found that the search augmentation makes a significantly higher positive contribution to the SAIL model than all other models. With `GPT-3.5-Turbo`, feeding search-augmented prompts with instructions leads to very slight improvements on both evaluations. However, grounding on search results can hurt the performance of Vicuna and LLaMA-GPT4 models of different sizes. By comparing against `GPT-4`, `Vicuna-13B` is slightly improved by search results, but the improvement is not present when compared to `GPT-3.5-Turbo`. For the `Vicuna-7B` and `LLaMA-7B-GPT4` baselines, augmenting input prompts with search engine outputs makes a significant, negative impact on both evaluations. On the other hand, applying search augmentation to `SAIL-7B` significantly improves model performance on both experiments (84% to 90% and 98% to 103%).

On `New-Question-80`, we find that `GPT-4` and `GPT-3.5-Turbo` perform similarly and are both outperformed by `SAIL-7B`. There are two main challenges that cause this result. Firstly, the models need to reason based on new texts that never

appreared in their pretraining or fine-tuning corpora. Secondly, the texts provided by the search engines are noisy. On the other hand, SAIL-7B deals with the problems more successfully through search augmented instruction tuning. These results inform our findings:

- The search results contain useful information that can improve the performance of instruction-following language models.
- Without search-augmented fine-tuning, it is difficult for a language model to utilize valuable information among the complicated search results, and distracting retrieval results can mislead the generations. Generation on latest questions further signifies the challenge.
- Search-augmented instruction learning can help the model better utilize the valuable information among noisy search results and improve instruction-following performance.

### 3.2 Question Answering

**Common Knowledge.** The experiment results of question answering are shown in Table 1. CSQA, OBQA, and ARC-Challenge are open-ended, selection-based question-answering tasks. We compare instruction-tuned Vicuna-7B, Vicuna-13B, LLaMA-7B-GPT4, and SAIL-7B models under no-augmentation and search-grounded settings with different sources. All evaluations are zero-shot and instruction guided. Traditionally, a knowledgeable LLM can answer questions and select the most coherent and appropriate answers without external information. In each task, we want to evaluate the performance of different models and knowledge bases. We search Wikipedia (Wiki) with the BM25 retriever, and the web with DuckDuckGO (DDG), feeding the LLMs with the top-3 search results, which could contain unrelated and distracting information.

In general, we found that DuckDuckGo (DDG) leads to better performance for all models on all tasks because it is more flexible, covering a much wider range of information. This suggests the effectiveness of search engines over retrieving a static knowledge base. We found that both LLaMA and Vicuna-7B models can be slightly improved when search results are provided on most tasks. However, the overall performance is limited. The average accuracy of searched-augmented LLaMA-7B and Vicuna-7B is below 50%.

With Vicuna-13B, which is a roughly two times larger model, we get the best average performance (51.0%) on the three tasks without grounding information. However, adding search results hurts its accuracy in most experiments. While augmenting the model with DDG search results slightly improves the performance on CSQA and OBQA, the accuracy on ARC-Challenge is decreased by 1.4%. With BM25-based Wikipedia search results, the accuracy can decrease by as much as 1.8%. While the Vicuna-13B model achieves strong non-augmented performance, it is challenging to further improve the accuracy by utilizing helpful information in the search results.

In contrast, the SAIL-7B model improves on all tasks when incorporating the search results, and also achieves strong non-augmented performance. Without retrieval results, SAIL-7B significantly outperforms LLaMA and Vicuna-7B on all tasks with a large margin (49.5% vs 44.5% and 40.9% average accuracy). It also performs slightly better than Vicuna-13B on CSQA and OBQA tasks, while Vicuna-13B is still strongest on ARC-C. While search augmentation leads to at most 0.5% improvement for Vicuna-13B, DDG search results improve SAIL-7B by 2.8% on OBQA and 1.2% on average, showing that the SAIL-7B model can steadily utilize the helpful information among the search results. As a result, the search-augmented SAIL-7B model achieves the best performance on both CSQA and OBQA.

**TruthfulQA.** We use the TruthfulQA evaluation set (Lin et al., 2022) containing 817 questions to evaluate how informative and truthful are language models. Following the standard approach, we fine-tuned a GPT-3 model for automatic evaluation. We compare the plain LLaMA models and search augmented instruction following models using the GPT-3 evaluator. The performance of different models and settings are shown in Table 2.

We notice that with search grounding, both truth and info scores can be significantly improved. By connecting with a search engine, the 7B models can significantly outperform the largest LLaMA-65B model. Similar to the automatic instruction-following results, the search-augmented SAIL-7B model outperforms search-augmented Vicuna-13B.

| Model | LLaMA-GPT4-7B | | | Vicuna-7B | | | Vicuna-13B | | | SAIL-7B | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Search | None | Wiki | DDG | None | Wiki | DDG | None | Wiki | DDG | None | Wiki | DDG |
| CSQA | 48.4 | 47.7 | 49.6 | 44.9 | 45.6 | 47.6 | 50.6 | 51.1 | 50.9 | 51.5 | 51.0 | **51.8** |
| OBQA | 42.2 | 44.4 | 44.6 | 37.2 | 39.4 | 42.6 | 49.0 | 47.2 | 49.4 | 49.2 | 50.2 | **52.0** |
| ARC-C | 43.0 | 45.2 | 47.3 | 40.5 | 44.5 | 46.3 | **53.2** | 51.6 | 51.8 | 47.7 | 48.1 | 48.4 |
| Avg. | 44.5 | 45.8 | 47.2 | 40.9 | 43.3 | 45.5 | **51.0** | 50.0 | 50.7 | 49.5 | 49.8 | 50.7 |
| Search Effect | none | +1.3 | +2.7 | none | +2.4 | +4.6 | none | -1.0 | -0.3 | none | +0.3 | + 1.2 |

Table 1: Question answering accuracy (%) by zero-shot models with simple instructions.

| Model | Size | True | True * Info |
|---|---|---|---|
| No Search Augmentation | | | |
| GPT-3 | 175B | 0.28 | 0.25 |
| LLaMA | 65B | 0.57 | 0.53 |
| LLaMA | 7B | 0.33 | 0.29 |
| Alpaca | 7B | 0.33 | 0.33 |
| Vicuna | 7B | 0.56 | 0.52 |
| W/ Search Augmentation | | | |
| Vicuna | 13B | 0.71 | 0.69 |
| Vicuna | 7B | 0.68 | 0.65 |
| SAIL | 7B | **0.73** | **0.73** |

Table 2: Automatic evaluation results of large language models on the TruthfulQA benchmark.

### 3.3 Fact and Fairness Checking

The other task we evaluate model performance on is unified fact and fairness checking (Zhang et al., 2023), a combined benchmark with four sub-tasks including fact-checking (Diggelmann et al., 2020; Kotonya and Toni, 2020), hate speech detection (de Gibert et al., 2018), and stereotype recognition (Sap et al., 2020). We evaluate the zero-shot performance on all four tasks, and the experiment results are shown in Table 3. The `SAIL-7B` model achieves the highest accuracy and F1 scores on all tasks, despite no grounding information being provided for the fact-checking tasks. We also found that the `Vicuna-7B` and `13B` models perform similarly on fact and fairness checking.

For the fact-checking tasks, we further evaluate the performance grounding on search results generated by DuckDuckGo. Grounding on an external search engine has both advantages and disadvantages. Many fact checking benchmarks provide task-specific grounding corpora that limit the domain of information retrieval. However, internet misinformation can be very arbitrary and related to the latest facts. A commercial search engine is able to catch a wide range of up-to-date information that a retrieval model with a fixed knowledge base cannot achieve. However, search engines are usually less accurate than dense retrievers, and they might retrieve disputed documents that influence the quality of fact checking. Our experiments show that the search results are not helpful for all baseline models. On Climate-Fever, augmenting the model with search results decreases the overall accuracy of `LLaMA` by 3%. On the PubHealth task, both accuracy and F1 of `Vicuna-13B` model are decreased by the search results, by 4% and 1% respectively. This shows that the search results contain distracting information, which prevents the models to utilize helpful evidence among noises.

However, SAIL is more robust against distracting languages and its fact-checking performance is improved on the same set of search results, as shown in Table 4. With search augmentation, the fact-checking accuracy and F1 scores of SAIL are improved on both tasks, as high as 4.2% on Climate-Fever. The augmented SAIL model also significantly outperforms all baselines, including `Vicuna-13B` and `LLaMA-7B` tuned with GPT-4 responses by 9% accuracy and 5% F1, showing the effectiveness of search augmented fine-tuning.

## 4 Related Work

**Large language models.** Beginning with GPT-3 (Brown et al., 2020a), LLMs have demonstrated strong abilities in knowledge memorization and text-based inference on a wide range of tasks. Well-known LLMs include GPT-3, LaMDA (Thoppilan et al., 2022), FLAN (Wei et al., 2021), OPT (Zhang et al., 2022), and LLaMA (Touvron et al., 2023). Compared to smaller language models, LLMs have several emergent abilities (Wei et al., 2022a), including zero-shot multi-task solving, and few-shot in-context learning with chain-of-thought reasoning (Wei et al., 2022b;

| Model | Metric | Climate | PubHealth | Fact Avg. | HSD | SBIC | Fairness Avg. | All Avg. |
|---|---|---|---|---|---|---|---|---|
| Vicuna-7B | Acc | 57.9 | 60.6 | 59.2 | 55.9 | 74.5 | 65.2 | 62.2 |
| | F1 | 38.8 | 56.63 | 47.7 | 68.5 | 84.3 | 76.4 | 62.04 |
| Vicuna-13B | Acc | 51.4 | 54.4 | 52.9 | 57.7 | 72.3 | 65.0 | 59.0 |
| | F1 | 42.5 | 57.7 | 50.1 | 69.6 | 82.9 | 76.3 | 63.2 |
| LLaMA-7B | Acc | 58.8 | 59.9 | 59.3 | 62.3 | 74.8 | 68.6 | 63.9 |
| | F1 | 46.6 | 57.5 | 52.0 | 72.3 | 84.4 | 78.4 | 65.2 |
| SAIL-7B | Acc | **63.5** | **69.2** | **66.4** | **70.1** | **76.4** | **73.2** | **69.8** |
| | F1 | **51.0** | **63.6** | **57.3** | **75.1** | **83.9** | **79.5** | **68.4** |

Table 3: Instructed zero-shot language checking performance on the UniLC benchmark.

| Model | Metric | Climate | PubHealth | Avg. |
|---|---|---|---|---|
| Vicuna-7B | Acc | 57.7 | 60.1 | 58.9 |
| | Acc Diff | -0.2 | -0.5 | -0.3 |
| | F1 | 49.5 | 57.6 | 53.6 |
| | F1 Diff | +10.7 | +1.0 | +5.9 |
| Vicuna-13B | Acc | 53.5 | 50.3 | 51.9 |
| | Acc Diff | +2.1 | -4.1 | -1.0 |
| | F1 | 46.6 | 56.8 | 51.7 |
| | F1 Diff | +4.1 | -0.9 | +1.6 |
| LLaMA-7B | Acc | 55.8 | 62.8 | 59.3 |
| | Acc Diff | -3.0 | +2.9 | -0.1 |
| | F1 | 50.2 | 59.7 | 54.9 |
| | F1 Diff | +3.6 | +2.2 | +2.9 |
| SAIL-7B | Acc | **65.8** | **70.7** | **68.3** |
| | Acc Diff | +2.3 | +1.5 | +1.9 |
| | F1 | **55.2** | **64.5** | **59.9** |
| | F1 Diff | +4.2 | +0.9 | +2.5 |

Table 4: Search augmented zero-shot fact checking on the Climate-Fever and PubHealth benchmarks.

Wang et al., 2022a).

**Instruction following.** Pretrained LLMs can generate texts following certain formats and rules by seeing a few examples in their prompts. To make LLMs more scalable and improve zero-shot performance, Ouyang et al. (2022) proposed training GPT-3 with instruction-response corpora. As a result, InstructGPT, ChatGPT, and GPT-4 can handle a wide range of tasks without seeing any examples. Recent research has also found that both GPT-generated instructions and instruct-following outputs (Peng et al., 2023) can improve the instruction-following ability of LLMs. (Wang et al., 2022a) proposed a semi-supervised method to generate diverse instructions based on a seed instruction base on NLP tasks (Mishra et al., 2022; Wang et al., 2022b). A more recent study shows that GPT-4 (OpenAI, 2023) can generate high-quality instruction-following language. Recent efforts on open-sourcing instruction-following LLMs include Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023).

**Retrieval-augmented language models.** Prior to our work, several initiatives explored retrieval-augmented language models (RALMs). The pioneering approaches – REALM (Guu et al., 2020) and RAG (Lewis et al., 2020) – sought to train language models with retrievers in an end-to-end manner. RETRO (Borgeaud et al., 2022) introduced the idea of training an LM on top of a frozen retriever. Atlas (Izacard et al., 2022) further explored dedicated loss functions for the end-to-end training of the retriever and the LM, achieving superior performance on several few-shot learning tasks. Recently, RePlug (Shi et al., 2023) and In-context RALM (Ram et al., 2023) instead explore an opposite direction: use a frozen black-box LM while fine-tuning the retrieval modules. RePlug shows its advantages of leveraging large LMs like Codex (Chen et al., 2021) and GPT-3 (Brown et al., 2020b), outperforming Altas on few-shot question-answering tasks.

Despite the success of RALMs, most of these models have limitations, including 1) constraining the search space to a closed corpus like Wikipedia 2) lacking explicit mechanisms for disregarding distracting search results, and 3) applying a few-shot in-context learning setting without considering instruction fine-tuning during RALM training. Consequently, their applications remain relatively narrow, primarily focusing on tasks such as question-answering and language modeling. SAIL addresses these limitations by 1) employing real-world search engines, 2) introducing a search result denoising

process capable of filtering out distracting information, and 3) incorporating instruction fine-tuning. Consequently, SAIL demonstrates its superiority in broader applications, including instruction following for chatbots, fact and fairness checking, all of which benefit from access to up-to-date information retrieved from real-world search engines.

## 5 Conclusion

In this work, we found that disputed and distracting search results can significantly mislead the predictions of large language models. Several transparency-sensitive tasks, including open-domain question answering and language checking can be negatively influenced by this phenomenon. To solve this problem, we propose a search-augmented instruction-following large language model with 7B parameters. We construct the first search-augmented instruction-tuning corpus consisting of human-generated instructions, GPT-4 generated responses, and search results generated by a BM25 retriever based on Wikipedia and a commercial search engine. We then fine-tuned the `LLaMA-7B` language model with the constructed training corpus on academic computational resources. Experiments on instruction-following, question answering, and fact/fairness checking show that the search-augmented language model can distill trustworthy and helpful information from all search results and generate high-quality responses, improving both the performance and transparency of instruction-following large language models.

## Acknowledgement

This research was supported by the Center for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission's InnoHK Scheme.

## Limitations

While the model we propose achieves high performance with efficient model settings, the major limitation of the model is that it does not explain why a search result is trustworthy or informative or not. In future work, we will fine-tune larger models and enable the models to recognize trustworthy search results with explanations. In the context of automated instruction-following evaluation, we adhere to the established literature that employing GPT-4 as the evaluator (Peng et al., 2023; Chiang et al., 2023; Liu et al., 2023). The robust performance of GPT-4 on Question-80 benchmark shows its efficacy. However, on the New-Question-80 benchmark, GPT-4 exhibits a comparatively diminished level of performance due to the need of latest knowledge. While this may introduce an element of uncertainty into the evaluation process, we believe that presenting these results can encourage the research community to be more informed and interested in this challenging problem. We plan to incorporate a human-in-loop evaluation as part of our future endeavors, ensuring a more precise and comprehensive assessment of instruction-following capabilities of large language models.

## References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ona de Gibert, Naiara Perez, Aitor Garcıa-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *EMNLP 2018*, page 11.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614*.

Jiaxin Ge, Hongyin Luo, Yoon Kim, and James Glass. 2023. Entailment as robust self-learner. *arXiv preprint arXiv:2305.17197*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Hongyin Luo and James Glass. 2023. Logic against bias: Textual entailment mitigates stereotypical sentence reasoning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1243–1254, Dubrovnik, Croatia. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca

Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

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

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. 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`.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Interpretable unified language checking. *arXiv preprint arXiv:2304.03728*.

| **Models** | Vicuna-7B-v1.1 | SAIL-7B |
|---|---|---|
| **Novel Verbs** | Include<br>Consider | Calculate<br>Match<br>Revolutionize<br>Check<br>Include<br>Increase |
| **Count** | 2 | **6** |

Table 5: Top-10 verbs generated by LLaMA-based models that do not overlap with GPT-4 and ChatGPT.

| **Models** | **Avg.** | **Std.** | **Diversity** |
|---|---|---|---|
| GPT-4 | 303.8 | 121.5 | 0.48 |
| ChatGPT | 135.1 | 63.6 | 0.56 |
| Vicuna-13B | 204.1 | 82.9 | 0.45 |
| Vicuna-7B | 196.5 | 90.3 | 0.45 |
| SAIL-7B + Search | 246.2 | 87.7 | 0.44 |
| SAIL-7B | 206.6 | 86.9 | 0.47 |

Table 6: Statistics about the length and diversity of the generated responses of different language models. Diversity stands for the total number of different words divided by the total length.

## A Data Statics

We first show the word preference of different models on the 80 unseen instructions. The results are shown in Figure 4. We compare the distributions of top-10 verbs generated by `GPT4`, `GPT-3.5-Turbo` (ChatGPT), `Vicuna-7B-v1.1`, and `SAIL-7B` models. With search augmentation, `SAIL-7B` generates significantly more verbs that do not overlap with GPT's generations, as shown in Table 5. Only two top-10 verbs generated by Vicuna are not covered by GPT-4 and ChatGPT, while six out of ten verbs generated by SAIL-7b are not high-frequency verbs by the GPT models. This indicates that the grounding search results can shift the generation preference of the language models.

The statistics of the generated responses is shown in Table 6. GPT-4 generates the longest and most diverse responses, while ChatGPT tends to generate shorter and simpler answers. Without search augmentation, the lengths of SAIL-7B generated sequences are similar to the Vicuna models. This indicates that search augmentation can increase the length of the generated responses.

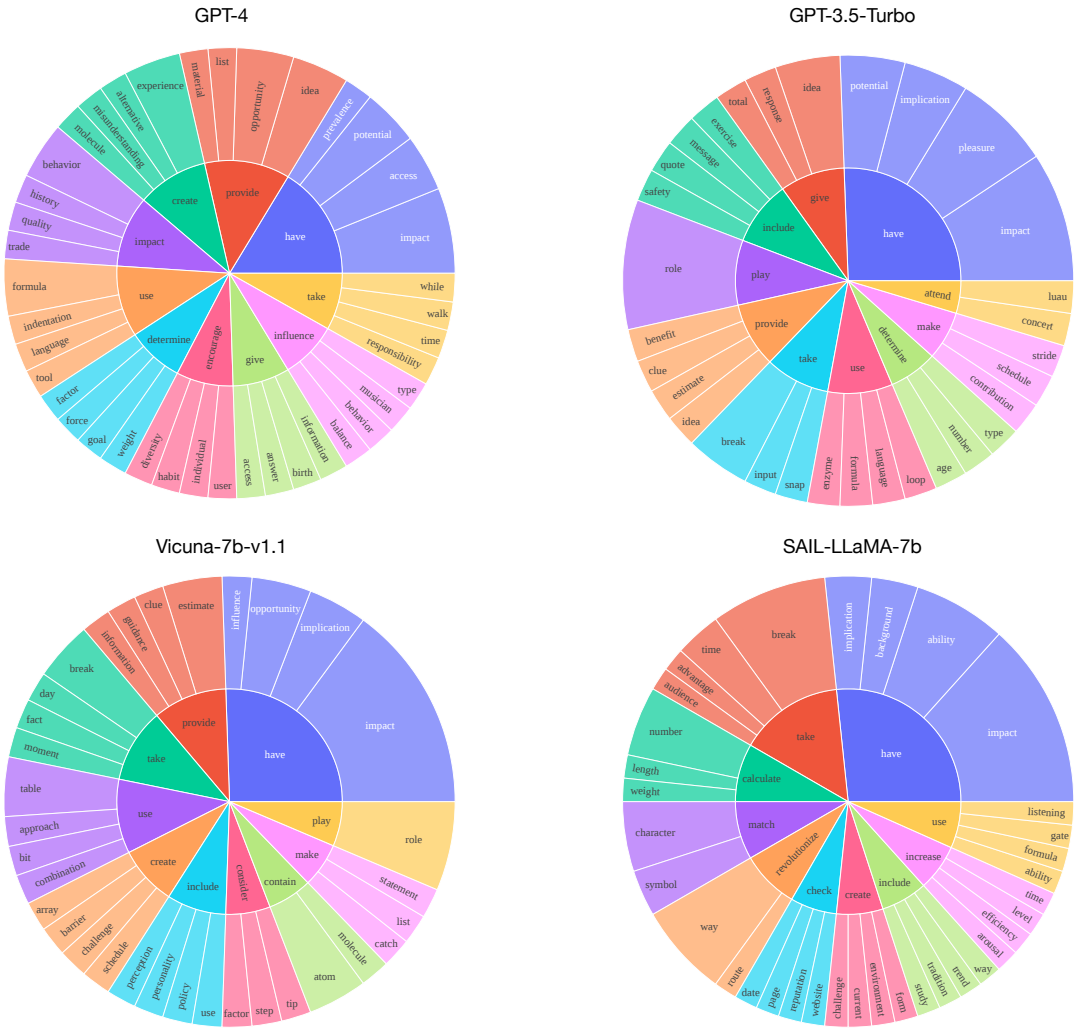


Figure 4: Top-10 verbs and associated nouns generated by selective large language models.