

# InstructSum: A Benchmark to Evaluate Instruction-Following Capability of Large Language Models in Summarization

Kosuke Nishida<sup>†</sup>, Kyosuke Nishida<sup>†</sup>, Itsumi Saito<sup>‡</sup>

<sup>†</sup> Human Informatics Labs., NTT, Inc., Japan

<sup>‡</sup> Tohoku University, Institute of Science Tokyo, RIKEN AIP, Japan  
{kosuke.nishida, kyosuke.nishida}@ntt.com, itsumi.saito@tohoku.ac.jp

## Abstract

Pre-trained large language models (LLMs) align their outputs with user intent through natural language instructions. In the summarization task, conciseness of the output is inherently required, which makes the instruction-following capability of LLMs particularly important. That is, providing supplementary information beyond the instruction can be undesirable. In this study, we introduce a novel benchmark, InstructSum, consisting of 3,309 types of instructions to evaluate the instruction-following capability in the summarization task. InstructSum has multiple instructions per source text, and thus it enables the evaluation of how LLMs adjust the content of the summary according to the instructions. Our experiments with six LLM families revealed the challenges that LLMs face in this task. For example, LLMs provide polite and helpful responses with irrelevant information; they go beyond instructions and fail to respond with a concise summary.

**Keywords:** Summarization, Evaluation Methodologies, Language Modelling

## 1. Introduction

Large language models (LLMs) control their output by referring to instructions or input and output examples in the prompt (Ouyang et al., 2022; Touvron et al., 2023a,b; Bai et al., 2023; Jiang et al., 2023; Gemma Team et al., 2024). Following natural language instructions is a fundamental capability of LLMs to meet users' requirements in their practical applications. While LLMs acquire this capability during post-training, they align their responses with human preferences (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). As a result, to respond helpfully, LLMs often provide redundant information beyond the instruction (Saito et al., 2023; Chen et al., 2024a; Miao et al., 2025).

Because summarization, a prominent application of natural language processing and LLMs (Hermann et al., 2015; See et al., 2017; Liu et al., 2024c), inherently requires a concise response, the instruction-following capability is a crucial problem; a verbose response exceeding the topic of the instruction is not appropriate as a summary. However, although general instruction-following capability has been extensively studied (Zhou et al., 2023; Qin et al., 2024a; Jiang et al., 2024a), the instruction-following in summarization has received limited attention.

Here, the goal of this paper is to enable the evaluation of the instruction-following capability in summarization with an emphasis on a perspective of conciseness. To this end, we provide a novel benchmark, InstructSum. The concept of Instruct-

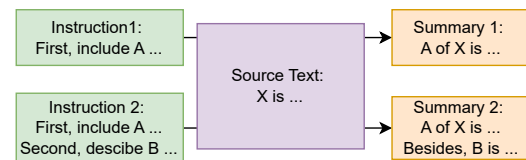


Figure 1: Concept of InstructSum. Each source text is paired with multiple instructions, and the instructions specify an increasing number of aspects. We evaluate LLMs' instruction-following capability by focusing on the consistency between the content in the summary and the aspects in the instruction.

Sum is illustrated in Figure 1. The novelty of InstructSum lies in its cumulative instruction structure, where each source text is associated with multiple instructions, and each successive instruction extends the previous one by incorporating additional aspects. LLMs are prompted to generate a summary as concise as possible that covers only the aspects specified in the given instruction. To create InstructSum, writing instructions places a heavy burden on annotators because it requires identifying multiple aspects in a summary and formulating an instruction for each aspect. Thus, to alleviate the burden, we made clusters of summaries written in a similar manner, then asked expert annotators to write the common manner in a cluster as the instruction. This clustering-based approach offers several advantages. First, writing instructions without guidance is open-ended and may lead to confusion. Clustering provides guidance that reduces confusion and improves efficiency. Second, clustering serves as data augmentation because a set of

This work was conducted while the third author was affiliated with NTT, Inc.

instructions can be used for multiple pairs of source and target text.

Beyond the dataset creation, we introduce two types of metrics for InstructSum. The first type uses the LLM-as-a-judge framework to evaluate the over-coverage and under-coverage of the generated summary relative to the given instruction. The second performs pairwise comparisons between summaries generated under the different instructions for the same source text.

We evaluated six LLM families with InstructSum. Experimental results revealed the challenges of LLMs in the summarization task. In particular, we observed that LLMs often failed to generate a concise summary. LLMs provided information beyond the instructions, even though they were instructed to generate a summary as concise as possible. This tendency was remarkable especially when the instruction consisted of one aspect.

## 2. InstructSum

First, we define the instructive summarization task in §2.1. Then, we describe the dataset creation process of the dataset satisfying the formulation in §2.2, §2.3, §2.4, and Figure 2.

### 2.1. Problem Formulation

**Instructive Summarization.** LLMs output the summary  $T$  when given an instruction  $I$  and source text  $S$ . We expect and guide LLMs to generate a summary for an instruction as concise as possible while it meets the requirements of the instruction.

### 2.2. Raw Data Collection

We first collected Wikipedia articles and create the target texts. Second, we generated the source text from the target text. Finally, we clustered the target texts for annotation efficiency by leveraging the edit distance of the Wikipedia titles and abstracts.

**Collection of target texts.** Following WikiSum (Liu et al., 2018), we used the abstract section of a Wikipedia article as the target text.<sup>1</sup>

**Generation of source texts.** Defining a source document for a Wikipedia abstract is difficult; the article body often does not contain information stated in the abstract. Thus, we generate a source text with gpt-oss-120b (Agarwal et al., 2025).<sup>2</sup> Figure 3

<sup>1</sup>We used the dump data of Wikipedia CirrusSearch on October 11th, 2022. <https://dumps.wikimedia.org/other/cirrussearch/>

<sup>2</sup>For gpt-oss-120b, we followed the default `reasoning-effort = medium` setting in all cases.

shows the prompt. We discuss the quality of the original and generated source texts in §2.5.

**Clustering of target texts.** We clustered the target texts by referring to Wu et al. (2021), which proposed a task to write a sketch of multiple Wikipedia abstracts. First, we grouped articles whose titles were identical except for a token at the same position. Then, we measured the Levenshtein distance of all article pairs in the cluster, and removed articles whose minimum distance was larger than 70% of its token length from the cluster. Finally, we deleted clusters with fewer than five members.

### 2.3. Dataset Annotation

Here, we created instruction sentences that is a part of the final instruction. The instruction sentences indicate common content in the target texts in a cluster. Thus, the resulting instruction sentences work as a guidance to generate a target text from the corresponding source text.

Then, we created the relation label between each instruction sentence and each sentence in a target text to clarify which sentence in a target text meets the requirements of an instruction sentence. Although this information is not required for evaluation, this process helps verify the quality of the instruction creation process. Also, it enables InstructSum to be used as the post-training dataset in future work.

**Annotation process.** We hired six professional annotators and supervised them with close communication and verification. All annotators were native English speakers. We assigned an annotator the task of writing instructions for a cluster to read five articles in the cluster. Then, another annotator constructed binary label matrices for all of the articles in that cluster. Each value in the binary matrix indicates whether a sentence in the article contains the content requested by an instruction sentence. Instead of hiring many crowd workers and assigning multiple crowd workers to the same cluster to measure the level of agreement, we hired and trained a small number of expert annotators in order to ensure the quality. Because we assigned an professional annotator to each cluster, we did not compute the annotator agreement. However, the labeling process reduced noise in the instruction annotation by verifying the relationship among all instructions and all target texts in a cluster. It took on average 20 minutes to write instructions for five articles and 72 minutes to label the sentences in all articles of a cluster.

**Annotation of instruction.** We asked the expert annotators to write instructions for 900 clusters and

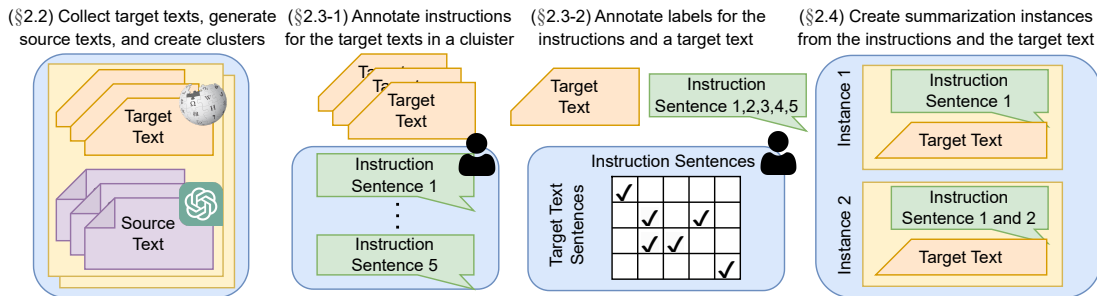


Figure 2: Data creation process of InstructSum. Objects in blue boxes are created in each step. In the first step, described in §2.2, we collected target texts from Wikipedia, generated source texts, and clustered them. In the second steps in §2.3, annotators wrote instruction sentences for a cluster so that the instruction sentences mentioned common content in the cluster. They labeled whether each sentence in a target text has information indicated by each instruction sentence. In the third step in §2.4, we created  $(I, S, T)$  instances by merging instruction sentences.

```

[Instruction]
You are a professional writer. Given a Wikipedia-style
summary and its topic, write a detailed article of 500 to 1000
words.
The article must include all information and key terms from
the summary, integrated naturally throughout the text and
spread across multiple paragraphs, not concentrated in the
first or last paragraph.
Do not restate the given summary in the first or last
paragraph, and do not create a leading or concluding
paragraph that summarizes the article itself.
You may add additional context, background, examples, and
related information to make the article rich and complex.
Your output must contain only the article itself, with no
explanations or instructions.

[Topic]
{Wikipedia title}

[Summary]
{target text}

[Article]

```

Figure 3: Prompt for source text generation.  $\{x\}$  means the substitution of  $x$ .

8,988 target texts. They wrote five instruction sentences per cluster according to the following procedure. To reduce the burden on the annotators, they read five target texts in a cluster.

Firstly, the annotators extracted keywords of a class (e.g., Donald Trump and Joe Biden) from each target text and noted the class name (President’s Name). Secondly, they wrote an instruction sentence mentioning a class name that is popular in the cluster. If the order of appearance of the class keywords was common throughout the target texts, the instruction sentence mentioned the order. If the number of target texts including the class keywords was less than four, it mentioned the condition under which to include them. The annotators were given five source texts per cluster, and thus we consider that class keywords appearing in five or four source texts are frequent information in the cluster. In con-

	#Items	Token Length
Cluster	900	—
Instruction	900	65.2
Source Text	8988	1065.3
Target Text	8988	91.7

Table 1: Statistics of the raw format version of InstructSum, which is before the transformation into the instructive summarization format. The length are counted with spaCy (Honnibal and Montani, 2018) `en-core-web-sm` tokenizer and averaged among the dataset. Each instruction has five instruction sentences.

trast, class keywords appeared in only three or two source texts are rare information in the cluster, and we asked the annotators to specify the cases in which the target text includes such information.

**Annotation of relation between instruction and target text.** The annotators annotated a binary matrix per target text. The cells in the matrix indicated whether each target sentence, which was split from the target text, included the content mentioned in each instruction sentence. The motivations of this procedure is not only to collect the labels but also to secure and feedback annotation quality.

**Data statistics.** Table 1 shows the data statistics of the annotated datasets. We provide histograms of token lengths in Appendix A.

## 2.4. Transformation to the instructive summarization task

Then, we transformed the dataset to the format defined in §2.1. From the instructions consisting

of five sentences, we created five instructions  $I$  by concatenating the instruction sentences from 1 to  $j$  ( $j = 1, \dots, 5$ ). Because we use LLM-as-a-judge evaluation, instead of modifying the target text according to the corresponding instruction, we instructed the LLMs that the target text was written independently of the instruction. We removed any instructions that were linked to non-successive sentences in the target text to avoid impractical instructions. Note that we denote the instructions described in this subsection as  $I$ ; that is,  $I$  are not the raw data collected and annotated in the previous subsection.

## 2.5. Dataset Analysis

**Data statistics.** Table 2 shows the data statistics.  $L_I$ ,  $L_S$ , and  $L_T$  denote the token lengths of the instruction, source text, and target text, respectively.

**Quality of instruction.** Figure 4 shows the distribution of first words of the 4,500 instruction sentences. 13.0% of the sentences are conditional instructions beginning with “If.” An example is “If the song was included on the greatest hits album, write it down including the song name.” 36.7% of the sentences refer to the order of appearance of the class keywords by beginning with specific adverbs, such as “First” and “Next.” These instructions are more complex than those that specify only a keyword class. The remaining 50.3% specify a keyword class with a variety of expressions.

**Quality of source text.** Because the source text is generated by gpt-oss-120b, there is a concern about the quality of the source text. During generation, we conducted manual spot checks on two aspects: the information included in the target text must be covered by the source text and appear across its multiple parts. We found that the issue with coverage was minimal. However, the information dispersion requires greater attention in the pilot version of our dataset because the model often inserted the target text unchanged into the opening paragraph of the source text. Thus, we added corresponding constraints (e.g., “integrated naturally throughout the text and spread across multiple paragraphs”), to the prompt shown in Figure 3.

As a result of such prompt engineering, our spot checks suggested that the source texts were aligned with the targets. Table 3 shows an example of a source text. We observed that the information in the target text is covered and appeared across multiple parts in the source text. Also, we calculated the  $n$ -gram coverage of the target text by the source text. We compared the coverage in InstructSum with two summarization datasets, CNN/Daily Mail (CNN/DM) (Hermann et al., 2015) and Multi-

$\#S$	$\#I$	$\#T$	$L_S$	$L_I$	$L_T$
8,993	3,309	8,993	1065.3	65.2	91.7

Table 2: Statistics of InstructSum in the instructive summarization format. The token length of the summary ( $L_S$ ), instruction ( $L_I$ ), and target text ( $L_T$ ) is averaged over the dataset.

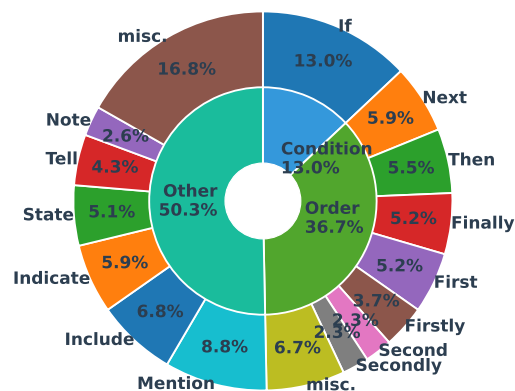


Figure 4: Classification of instructions and distribution of first words.

News (Fabbri et al., 2019). Here, we reference CNN/DM and Multi-News because they are commonly used benchmarks and the lengths of their source and target texts are similar to ours. Table 4 lists the results. The  $n$ -gram coverage is comparable, which indicates that the source texts in InstructSum are consistent with the target texts as in the standard datasets. Also, we found that increasing  $n$  decreases the coverage more in InstructSum than the other datasets. We consider that this is because our prompt instructing “You are a professional writer” led to frequent paraphrasing of key phrases in the target text. Furthermore, we confirmed that generating source text leads to higher  $n$ -gram coverage than using the original source text (i.e., the Wikipedia body).

## 2.6. Data Distribution and Data Split

Table 5 shows the distribution of the categories of the Wikipedia articles in the dataset. The categories were automatically classified by CirrusSearch. We note that an article and a cluster may belong to multiple categories.

In order to evaluate the control capability with regard to unseen instructions, we used 104 clusters that did not include Geography articles as the test set. The remaining 796 clusters were split randomly into 746 training and 50 development clusters.

Title: 70th Primetime Creative Arts Emmy Awards
Target Text: The 70th Annual Primetime Creative Arts Emmy Awards ceremony were held over <b>two nights on September 8 and September 9, 2018</b> . <b>The nominations were announced on July 12, 2018</b> . The ceremony was <b>in conjunction with the annual Primetime Emmy Awards</b> and is presented in <b>recognition of technical and other similar achievements in American television programming, including guest acting roles</b> .
Source Text: The Creative Arts Emmy Awards have long served as the backstage engine that powers the glitz of primetime television, honoring the craftspeople whose work often goes unseen by the casual viewer. . . . Over the decades, the awards have evolved from <b>a modest adjunct to the main Primetime Emmys</b> into a sprawling, multi-night celebration that mirrors the growing complexity of modern television production. In 2018, the ceremony reached a new milestone with its 70th iteration, reflecting seven decades of industry innovation. <b>The nominations, unveiled on July 12, . . .</b> The awards themselves unfolded over <b>two consecutive evenings, September 8 and September 9, 2018, . . .</b> One of the most talked-about moments of the 70th Creative Arts Emmys was <b>the recognition of guest actors who delivered memorable, albeit brief, contributions to their respective series</b> . <b>The guest acting categories—spanning drama, comedy, and limited series—have become a barometer for the industry’s appreciation of talent that can elevate a show in just a handful of scenes. . . .</b>

Table 3: Example of a source and target text pair. The source text is omitted, which consists of 17 paragraphs and 822 words.

	1	2	3	4	$L_S$	$L_T$
InstructSum	77.3	37.8	17.2	8.61	1065.3	91.7
CNN/DM	82.7	46.5	27.9	19.1	785.7	66.7
Multi-News	72.0	39.6	23.7	17.7	2119.5	262.4
Original source	56.6	22.4	10.2	5.21	1744.1	91.7

Table 4: Percentage of  $n$ -gram coverage in the summarization datasets. The left columns indicates the uni-gram, bi-gram, tri-gram, and 4-gram coverage, respectively. The rest is the token length of the summary, and target text ( $L_T$ ).

### 3. Evaluation Metrics

To evaluate the LLM’s instruction-following capability in the summarization task, we introduce four metrics.

**Precision-GPT (P-GPT).** We evaluate the conciseness of summaries that should include the information requested by the instructions. We obtain this score by prompting the gpt-oss-120b model to rate each summary on a scale from 1 (verbose) to 10 (concise). We adopt the open-source gpt-oss-120b model for reproducibility and cost efficiency. Figure 5 shows the prompt. We report the average score over the test split  $\mathcal{D}_{\text{test}}$ .

**Recall-GPT (R-GPT).** We evaluate that completeness of summaries that should satisfy all the aspects in the instruction. We calculate this score in

Geography/Regions	6920
Culture/Sports	2808
Culture/Media	2157
Culture/Biography	1969
History and Society/Politics and Government	1458
Culture/Media	958
Geography/Geographical	833
STEM/STEM	629
Culture/Visual Arts	523
History and Society/Society	485
History and Society/Transportation	431

Table 5: Top-ten categories/subcategories of articles in InstructSum. CirrusSearch groups Wikipedia into four categories: Geography, Culture, STEM, and History and Society.

the same manner as P-GPT. 1 is an incomplete summary, and 10 is a complete one.

**F1-GPT (F-GPT).** We report F-GPT, the harmonic mean of P-GPT and R-GPT, as a balanced metric.

**Length Consistency (L-Cons).** In the instructive summarization task, the content in the generated summary  $f(S, I)$  is determined by the instruction  $I$ . From the perspective of conciseness, the amount of content in the summary should increase as the number of aspects specified in the instructions increases. As explained in §2.4, each source text has five instructions; the instruction  $I_j$  consists of the first  $j$  instruction sentences (*i.e.*, sentences 1 through  $j$ ). Here, because  $I_j \subsetneq I_k$  if  $j < k$ , the content in  $f(S, I_k)$  must be richer than in  $f(S, I_j)$ . Thus, we define length consistency as follows:

$$E_{\mathcal{D}_{\text{test}}} \left[ \frac{1}{4} \sum_{1 \leq j \leq 4} \delta \{l(f(S, I_{j+1})) > l(f(S, I_j))\} \right],$$

where  $l(\cdot)$  is the token length and  $\delta(\cdot)$  is an indicator function.

Liu et al. (2024b) conducted a meta-evaluation based on 100 samples in the instructive summarization, which includes variants of P-GPT and R-GPT. They reported that Irrelevant Information, which corresponds to P-GPT based on gpt-4-0314 (OpenAI, 2023), and Missing Information, which corresponds to R-GPT, have correlations of 0.738 and 0.949, respectively, with the human annotations in a system-level evaluation.

## 4. Experiments

We investigated the instruction-following capability of state-of-the-art LLMs in the summarization task. We note that the purpose of our experiments is not

[Instruction]  
Please act as an impartial judge and evaluate the quality of the summary responded by an AI assistant to the user instructions displayed below. Instead of the source text, your evaluation must rely the reference summary written by a human from the same source text, independent of the instructions. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must judge **whether the summary provides only the information requested by the instructions on a scale of 1 (verbose) to 10 (concise) how completely the summary provides all the information requested by the instructions on a scale of 1 (incomplete) to 10 (complete)** by strictly following this format: "[[rating]]", for example: "Rating: [[5]].".

[Instructions for Assistant]  
{instructions}

[The Start of Assistant's Summary]  
{generated summary}  
[The End of Assistant's Summary]

[The Start of Reference Summary]  
{reference summary}  
[The End of Reference Summary]

Figure 5: Prompt for the precision score (colored in red) and recall score (blue).

Summarize this article as concisely as possible, following the instructions below. Include all points mentioned in the instructions, and do not add anything beyond them. Only provide the summary, without any additional explanation.

[Instructions]  
{instructions}

[Title]  
{Wikipedia title}

[The Start of Article]  
{source text}  
[The End of Article]

[Summary]

Figure 6: Prompt for inference.

to rank LLMs against each other, but to discuss the current achievements and challenges of LLMs.

#### 4.1. Experimental settings

**Models.** We adapted six LLM families: Llama3 (Grattafiori et al., 2024), Llama4-Scout (Meta AI, 2025), gemma3 (Gemma Team et al., 2025), Qwen3 (Yang et al., 2025), GPT-5 (OpenAI, 2025).<sup>3</sup>, and gpt-oss-120b. We used the non-reasoning mode for all evaluation, except gpt-oss, which was selected for the self-evaluation scenario.

**Inference.** We fed the prompts shown in Figure 6 to the model and generated summaries in a zero-shot fashion. We used vllm (Kwon et al., 2023b) on

<sup>3</sup>We used the gpt-5-2025-08-07 shapshot In the API call, we set reasoning-effort minimal and verbosity low.

	Size	P-GPT	R-GPT	F-GPT	L-Cons
Llama3	8B	2.22	8.18	3.33	48.5
	70B	3.18	8.55	4.11	50.5
Llama4	109B	6.38	7.96	6.56	60.4
gemma3	4B	4.93	7.99	5.61	54.3
	27B	6.92	8.55	7.16	60.9
Qwen3	4B	6.86	8.10	6.88	61.6
	8B	6.97	8.31	7.08	64.3
	14B	7.17	8.58	7.37	63.6
	32B	<b>7.86</b>	8.61	<b>7.85</b>	<b>72.1</b>
GPT-5	nano	6.75	8.83	7.16	64.8
	mini	6.69	9.16	7.23	61.4
		7.20	<b>9.31</b>	7.67	64.5
gpt-oss	120B	7.08	9.30	7.59	58.5

Table 6: Performance of the models.

eight NVIDIA H100 (80GB) GPUs for generation.

#### 4.2. Results and discussion

**Can LLMs generate concise summaries?** Table 6 shows the main results. We first focus on P-GPT, which measures the degree to which the content in the summary is required by the instructions. In other words, a smaller P-GPT means that the summary is more redundant with respect to the instructions. We found that the P-GPT scores were consistent with the model size except for GPT-5. This is reasonable because GPT-5 nano “is great for summarization and classification tasks” according to the official document.<sup>4</sup> The Qwen3-series showed strong performance even with the 4B model. The Llama3-series showed the lower P-GPT scores, indicating that it tends to provide polite responses with additional information rather than strictly following instructions.

#### Can LLMs generate informative summaries?

Secondly, we discuss R-GPT, which measures the degree to which the requirements in the instructions are satisfied in the summary. We found that the R-GPT performance remained consistently high across the models.

Comparing the P-GPT and R-GPT, all models had a higher R-GPT score than P-GPT. This tendency reflects the post-training strategy of LLMs. Because LLMs are trained to generate human-preferred responses, they tend to add potentially relevant information to minimize the risk of overlooking the given instructions. Consequently, LLMs often go beyond the given instruction and fail to produce concise summaries.

<sup>4</sup><https://platform.openai.com/docs/models/gpt-5-nano>

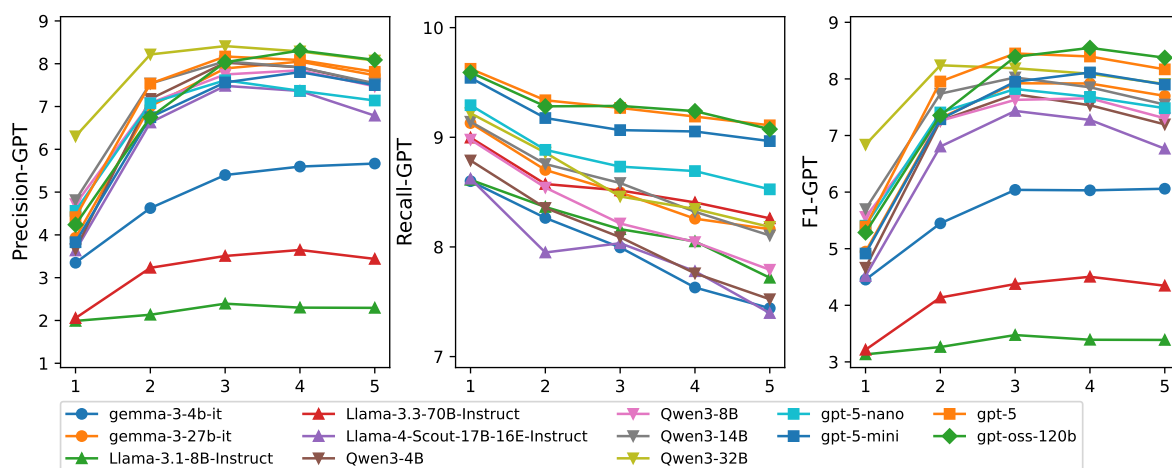


Figure 7: Average P-GPT, R-GPT, and F-GPT of generated summaries per number of instruction sentences.

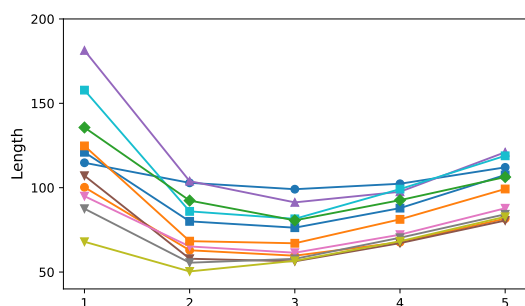


Figure 8: Average token length of generated summaries per number of instruction sentences. We omit the results of Llama3-series because they were over 700 tokens.

### Can LLMs generate balanced summaries?

Third, we mention F-GPT, the harmonic mean of P-GPT and R-GPT. We found consistency between F-GPT and model size, similar to the observation for P-GPT. Besides, Qwen3-series perform well due to their high P-GPT scores.

### Do LLMs adjust the amount of content in their summaries according to instructions?

Then, we explore L-Cons, which reflects the expectation that when an instruction requests more aspects, the content of the summary should increase accordingly. Considering that the chance level of L-Cons is 50%, we conclude that LLMs have only a limited ability to adjust the amount of content in response to instructions. We hypothesize that the LLMs learned a preferred generation length during post-training and tend to maintain this length even when instructed otherwise. To further advance this discussion, we pose the next research question.

### Can LLMs change the length of the summaries according to the instructions?

Figure 8 shows the average token length grouped by the number of instruction sentences. We observed two main trends: an overall upward trend for  $|I| \geq 3$  and the longest generations at  $|I| = 1$ . The upward trend is expected; more instructions are given, and the summarization becomes longer as the content required for the instructions increases.

We consider that the longest generation at  $\|I\| = 1$  reinforces the aforementioned observation that LLMs learn a preferred generation length. That is, LLMs tend to expand the output with additional information up to their preferred length, although they are instructed to generate summaries as concisely as possible. In contrast, if the instructions consist of multiple sub-queries, LLMs can determine what content to include in the summary based on the instructions, independently of their preferred length.

### Can LLMs change the contents of the summary according to the instruction?

Figure 7 shows the average P-GPT, R-GPT, and F-GPT grouped by the number of instruction sentences. P-GPT was defined such that 1 means verbose and 10 means concise. Therefore, P-GPT suffers from the same issue seen in the token length. At  $\|I\| = 1$ , the summary has redundant information to meet a certain token length, and the P-GPT scores worsen. When  $\|I\| \geq 2$ , the P-GPT scores exhibit an inverted U-shape, particularly when the LLMs achieve relatively high scores (e.g.,  $P\text{-GPT} \geq 6$ ). When more aspects are fed, it becomes more challenging to respond to all the requirements concisely. We consider this to be the reason for the inverted U-shape of P-GPT scores.

R-GPT was defined such that 1 means incomplete and 10 means complete. Thus, it showed an

Title: 70th Primetime Creative Arts Emmy Awards
Instructions: 1. State what the purpose of the awards are for. 2. Mention when the nominations were announced.
Summary for Instruction 1: (P-GPT 9, R-GPT 10) The purpose of the 70th Primetime Creative Arts Emmy Awards is to honor the technical and artistic contributions of craftspeople in television, including set design, costume creation, sound mixing, visual effects, guest acting, and other behind-the-scenes work that supports the production of primetime television.
Summary for Instruction 1 and 2: (P-GPT 9, R-GPT 10) The 70th Primetime Creative Arts Emmy Awards honor the technical and artistic contributions behind television production, including set design, sound, visual effects, and guest acting. The nominations were announced on July 12, 2018.

Table 7: First example of the generated summaries by Qwen3-32B.

overall downward trend, because as the number of aspects in the instruction increases, it becomes more challenging to satisfy all of them. The trend of F-GPT is consistent with that of P-GPT.

**Qualitative analysis.** We discuss the typical behaviors of the generated summaries in Instruct-Sum. Here, we focus on Qwen3-32B because it performed best in the F-GPT score. We list the first two instruction sentences and the corresponding generated summaries due to space limits.

Table 7 shows the first example. The corresponding source and target texts are shown in Table 3. We consider that the generated summary for instruction 1 is one of the expected behaviors of LLMs in the instructive summarization. The topics in the summary remained within the scope of the instruction (the purpose of the awards). However, the model helpfully added details within the topic to avoid a too simple response that humans do not prefer. The generated summary for instruction 1 and 2 is also acceptable. However, we found that the LLMs tend to answer the request in the first instruction extensively and to add a brief answer to the second instruction at the end. Such an imbalanced explanation problem, where the information regarding the second instruction is shorter and simpler than that of the first instruction, was observed throughout the experiments. This finding can be regarded as a novel form of positional bias (Liu et al., 2024a; Chen et al., 2024b; Zeng et al., 2025). The proposed L-Cons metric detects the imbalanced explanation problem by penalizing shorter summaries for more requests.

Table 8 shows the second example. The generated summary for instruction 1 is a typical error. It went beyond the topic of the instruction (universe), such as the season. This is because LLMs adhere to a preferred generation length and thus expand summaries with irrelevant information to reach a learned length. The P-GPT score detects this type

Title: Absolution (Agents of S.H.I.E.L.D.)
Instructions: 1. Include which universe the show is set in. 2. Write down the name of the director.
Summary for Instruction 1: (P-GPT 3, R-GPT 10) "Absolution" is part of the third season, which intertwines the aftermath of "Captain America: Civil War" with the series' own storyline. The episode, directed by Billy Gierhart and written by Chris Dingess and Drew Z. Greenberg, features the team battling Hive, an ancient Inhuman parasite, in a high-stakes conflict that tests their leadership, loyalty, and morality. Clark Gregg reprises his role as Phil Coulson, while Chloe Bennet, Brett Dalton, Ming-Na Wen, Iain De Caestecker, Elizabeth Henstridge, Henry Simmons, and Luke Mitchell deliver key performances. The episode aired on May 17, 2016, and was praised for its character development, visual style, and integration of MCU themes.
Summary for Instruction 1 and 2: (P-GPT 10, R-GPT 10) The episode "Absolution" is part of the "Agents of S.H.I.E.L.D." series set in the Marvel Cinematic Universe. It was directed by Billy Gierhart.

Table 8: Second example of the generated summaries by Qwen3-32B.

of error. The generated summary for instruction 1 and 2 is a simple yet expected example as the summarization output. However, the generated summary to the first instruction in the first example exhibits a polite and helpful style that is typical of LLMs' responses. We consider that both scores of 9 and 10 in P-GPT are desirable, and they capture different styles of summaries.

## 5. Related Work

### 5.1. Instruction-following capability of large language models

In the early days of the LLM era, the instruction-following capability of LLMs was fine-tuned and evaluated by providing a wide range of NLP tasks and viewing their task definition as the instructions (Wei et al., 2022; Wang et al., 2022). As the use of LLM has become widely spread and increasingly diverse, the fine-grained instruction-following capability has attracted attention. Due to the increased attention, some papers have released benchmarks that include instructions regarding various aspects in addition to the main task description (Zhou et al., 2023; Qin et al., 2024b; Jiang et al., 2024b; Chen et al., 2024b; He et al., 2024; Wen et al., 2024; Dussolle et al., 2025; Pyatkin et al., 2025). However, in general NLP tasks, a helpful yet verbose response can be acceptable. In summarization, where conciseness is essential, such verbosity is a crucial problem. Thus, we extended this line of work to summarization and created Instruct-Sum by focusing on the summaries for the different instructions and the same source text. In the instructive summarization task, Skopek et al. (2023); Liu et al. (2024b) investigated a meta-evaluation of this task and discussed potential evaluation met-

rics. Based on their meta-evaluation, we adopted the P-GPT and R-GPT metrics for InstructSum and conducted large-scale experiments on the various LLMs.

## 5.2. Controllable abstractive summarization

Abstractive summarization has been controlled by given queries (Dang, 2005, 2006; Nema et al., 2017; Kulkarni et al., 2020; Zhong et al., 2021; Zhu et al., 2022; Huang et al., 2023; Ramirez-Orta et al., 2025), aspects (Wang and Ling, 2016; Yang et al., 2018; Hayashi et al., 2021; T.y.s.s. et al., 2024; Zhong and Litman, 2025), keywords (Saito et al., 2020b; He et al., 2021; Liu and Chen, 2022), styles (Goyal et al., 2022; Bao et al., 2023), and lengths (Kikuchi et al., 2016; Saito et al., 2020a; Liu et al., 2022; Kwon et al., 2023a; Retkowski and Waibel, 2025). Although each specification method has its own advantages, InstructSum is novel because it focuses on controlling both the amount and the categories of information included in the summary via multiple textual instructions, and enables the evaluation of LLMs’ capabilities from this perspective.

## 5.3. Conciseness in LLM-based summarization

Reinforcement learning used to train LLMs can induce length bias (Shen et al., 2023; Singhal et al., 2024), which encourages verbose outputs. As LLMs that favor longer responses continue to be released, conciseness remains a fundamental principle of summarization and addressing it is increasingly important. Several recent studies have incorporated conciseness as an evaluation dimension (Liu et al., 2024b; Song et al., 2024; Lee et al., 2024; Song et al., 2025). This work highlights the importance of conciseness in LLM-based summarization by providing the evaluation framework that explicitly controls the amount of information in the summary.

## 6. Conclusion

To evaluate the instruction-following capability of LLMs in the summarization task, we introduced the InstructSum dataset and the evaluation metrics.

Our experiments clarified the following challenges of the state-of-the-art LLMs: (i) LLMs prefer the coverage of summaries to the conciseness; (ii) both conciseness and coverage increase as the model size grows; (iii) LLMs may learn a preferable token length of the response during their post-training, and they expand the summary until the length even if it beyond user’s intent; (iv) LLMs

heavily focused the first instruction and disregard later instructions. We believe that our InstructSum dataset and findings provide a solid foundation in exploration of the instruction-following capability in summarization.

## 7. Bibliographical References

- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Guangsheng Bao, Zebin Ou, and Yue Zhang. 2023. [GEMINI: Controlling the sentence-level summary style in abstractive text summarization](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 831–842.
- Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024a. [ODIN: Disentangled reward mitigates hacking in RLHF](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7935–7952.
- Xinyi Chen, Baohao Liao, Jirui Qi, Panagiotis Eustratiadis, Christof Monz, Arianna Bisazza, and Maarten de Rijke. 2024b. [The SIFo benchmark: Investigating the sequential instruction following ability of large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1691–1706.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. [Deep reinforcement learning from human preferences](#). In *Advances in Neural Information Processing Systems*, volume 30.
- Hoa Trang Dang. 2005. Overview of duc 2005. In *DUC@EMNLP*, volume 2005, pages 1–12.
- Hoa Trang Dang. 2006. [DUC 2005: Evaluation of question-focused summarization systems](#). In *Task-Focused Summarization and Question Answering@ACL*, pages 48–55.
- Antoine Dussolle, A. Cardeña, Shota Sato, and Peter Devine. 2025. [M-IFEval: Multilingual](#)

- instruction-following evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6161–6176.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Tanya Goyal, Nazneen Rajani, Wenhao Liu, and Wojciech Kryscinski. 2022. [HydraSum: Disentangling style features in text summarization with multi-decoder models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 464–479.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. [WikiAsp: A Dataset for Multi-domain Aspect-based Summarization](#). *TACL*, 9:211–225.
- Junxian He, Wojciech Maciej Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2021. [CTRLsum: Towards generic controllable text summarization](#). In *ICLR*.
- Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, et al. 2024. [Multi-if: Benchmarking llms on multi-turn and multilingual instructions following](#). *arXiv preprint arXiv:2410.15553*.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, page 1693–1701.
- Matthew Honnibal and Ines Montani. 2018. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *to appear*.
- Shaoyao Huang, Luozheng Qin, and Ziqiang Cao. 2023. [Diffusion language model with query-document relevance for query-focused summarization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11020–11030.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024a. [FollowBench: A multi-level fine-grained constraints following benchmark for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688.
- Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024b. [FollowBench: A multi-level fine-grained constraints following benchmark for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. [Controlling output length in neural encoder-decoders](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.
- Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. AQuaMuSe: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.
- Jingun Kwon, Hidetaka Kamigaito, and Manabu Okumura. 2023a. [Abstractive document summarization with summary-length prediction](#). In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 618–624.

- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023b. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Yuhoo Lee, Taewon Yun, Jason Cai, Hang Su, and Hwanjun Song. 2024. UniSumEval: Towards unified, fine-grained, multi-dimensional summarization evaluation for LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3941–3960.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *ICLR*.
- Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024b. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4481–4501. Association for Computational Linguistics.
- Yixin Liu, Kejian Shi, Katherine He, Longtian Ye, Alexander Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2024c. On learning to summarize with large language models as references. 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 8647–8664.
- Yizhu Liu, Qi Jia, and Kenny Zhu. 2022. Length control in abstractive summarization by pretraining information selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6885–6895.
- Zhengyuan Liu and Nancy Chen. 2022. Entity-based de-noising modeling for controllable dialogue summarization. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 407–418.
- Meta AI. 2025. The llama 4 herd of models.
- Yuchun Miao, Sen Zhang, Liang Ding, Yuqi Zhang, Lefei Zhang, and Dacheng Tao. 2025. The energy loss phenomenon in RLHF: A new perspective on mitigating reward hacking. In *Forty-second International Conference on Machine Learning*.
- Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072.
- OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- OpenAI. 2025. Introducing GPT-5.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. 2025. Generalizing verifiable instruction following. *arXiv preprint arXiv:2507.02833*.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024a. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024b. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048.
- Juan Ramirez-Orta, Ana Maguitman, Axel J. Soto, and Evangelos Milios. 2025. ROUGE-SciQFS: A ROUGE-based method to automatically create datasets for scientific query-focused summarization. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2187–2197.

- Fabian Retkowsky and Alexander Waibel. 2025. [Zero-shot strategies for length-controllable summarization](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 551–572.
- Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. 2020a. Length-controllable abstractive summarization by guiding with summary prototype. *arXiv preprint arXiv:2001.07331*.
- Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020b. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *arXiv preprint arXiv:2003.13028*.
- Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. [Verbosity bias in preference labeling by large language models](#). In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *ACL*, pages 1073–1083.
- Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. [Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2859–2873.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. [A long way to go: Investigating length correlations in RLHF](#). In *First Conference on Language Modeling*.
- Ondrej Skopec, Rahul Aralikkatte, Sian Gooding, and Victor Carbune. 2023. [Towards better evaluation of instruction-following: A case-study in summarization](#). In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 221–237.
- Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. [FineSurE: Fine-grained summarization evaluation using LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922.
- Hwanjun Song, Taewon Yun, Yuho Lee, Jihwan Oh, Gihun Lee, Jason Cai, and Hang Su. 2025. [Learning to summarize from LLM-generated feedback](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 835–857.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. [Learning to summarize with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Santosh T.y.s.s., Mahmoud Aly, and Matthias Grabmair. 2024. [LexAbSumm: Aspect-based summarization of legal decisions](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10422–10431.
- Lu Wang and Wang Ling. 2016. [Neural network-based abstract generation for opinions and arguments](#). In *NAACL*, pages 47–57.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoor-molabashi, 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, Krma Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 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.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. [Finetuned language models are zero-shot learners](#). In *ICLR*.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. [Benchmarking complex instruction-following with multiple constraints composition](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 137610–137645. Curran Associates, Inc.

Zequiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, and Bill Dolan. 2021. [Automatic document sketching: Generating drafts from analogous texts](#). In *ACL-IJCNLP (Findings)*, pages 2102–2113.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Min Yang, Qiang Qu, Ying Shen, Qiao Liu, Wei Zhao, and Jia Zhu. 2018. [Aspect and sentiment aware abstractive review summarization](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1110–1120.

Jie Zeng, Qianyu He, Qingyu Ren, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. 2025. [Order matters: Investigate the position bias in multi-constraint instruction following](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12479–12492.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [QMSum: A new benchmark for query-based multi-domain meeting summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921.

Yang Zhong and Diane Litman. 2025. [From information to insight: Leveraging LLMs for open aspect-based educational summarization](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1914–1947.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *arXiv preprint arXiv:2311.07911*.

Haichao Zhu, Li Dong, Furu Wei, Bing Qin, and Ting Liu. 2022. [Transforming wikipedia into augmented data for query-focused summarization](#). *TASPL*, 30:2357–2367.

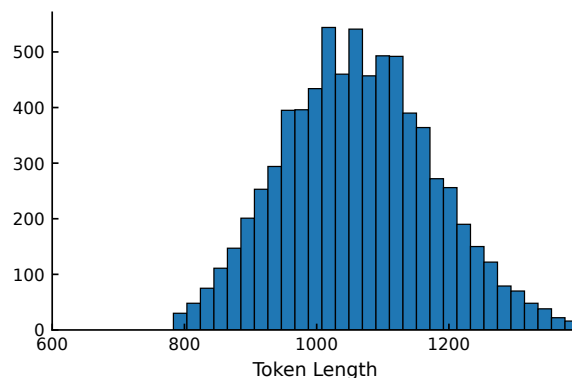


Figure 9: Histogram of token lengths for the source text.

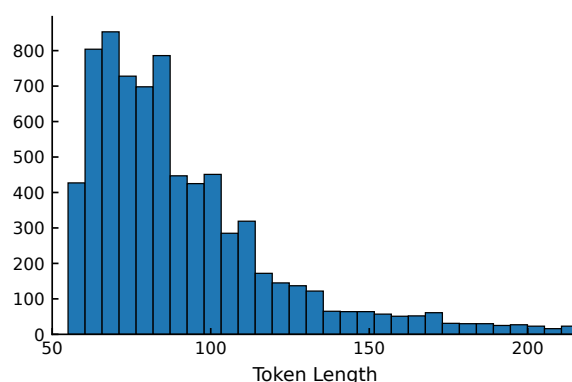


Figure 10: Histogram of token lengths for the target text.

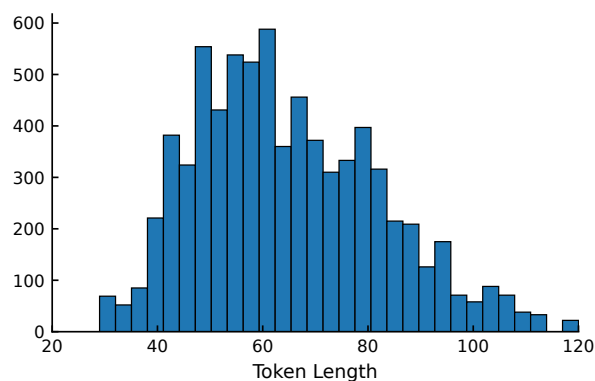


Figure 11: Histogram of token lengths for the instructions.

## A. Dataset Details

Figures 9, 10, and 11 show the histograms of the token lengths for the source text, the target text, and the instruction in InstructSum, respectively. We exclude values below the 0.5th percentile and above the 99.5th percentile for clarity.