SEvenLLM: Benchmarking, Eliciting, and Enhancing Abilities of Large Language Models in Cyber Threat Intelligence

Anonymous ACL submission

Abstract

To address the increasing complexity and frequency of cybersecurity incidents emphasized by the recent cybersecurity threat reports with billions of instances, cyber threat intelligence (CTI) plays a critical role in the modern cybersecurity landscape by offering the insights required to understand and combat the constantly evolving nature of cyber threats. Inspired by the powerful capability of large language models (LLMs) in handling complex tasks, in this paper, we introduce a framework to benchmark, elicit, and improve cybersecurity incident analysis and response abilities of **LLM**s for **S**ecurity **Even**ts (called **SEvenLLM**). Specifically, we create a high-quality bilingual instruction corpus by crawling cybersecurity raw text from cybersecurity websites to overcome the lack of effective data for information extraction. Then, we design a pipeline to auto-select tasks from the task pool and convert the raw text into supervised corpora comprised of question and response. The instruction dataset SEvenLLM-Instruct is used to train cybersecurity LLMs with the multi-task learning objective (28 well-designed tasks) for augmenting the analysis of cybersecurity events. Extensive experiments in our curated benchmark (SEvenLLM-Bench) demonstrate that SEvenLLM performs more sophisticated threat analysis and fortifies defenses against the evolving landscape of cyber threats.

1 Introduction

In the rapidly evolving landscape of cyberspace, the intricate and diverse nature of cybersecurity postures is undergoing exponential growth in complexity (Zhao et al., 2020; Arp et al., 2022; Alam, 2022). The digital realm is witnessing an unprecedented surge in cybersecurity incidents, with over 20 billion events impacting networks globally[1], which presents an immense challenge for cybersecurity professionals and analysts.

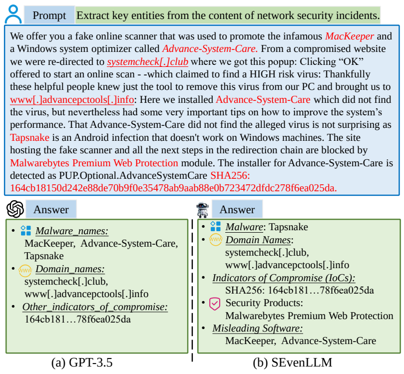


Figure 1: Comparison between GPT-3.5 and our proposed model SEvenLLM.

Cyber threat intelligence (CTI) (Liu and Zhan, 2023; Motlagh et al., 2024; Siracusano et al., 2023) is proposed to understand and anticipate cyber threats for cybersecurity defense. However, traditional Natural Language Processing (NLP) approaches often lack the requisite background knowledge for effective investigation. Large language models (LLMs) (Schick and Schütze, 2021; Achiam et al., 2023) present a groundbreaking shift in the capabilities of understanding and generating language-based content. Domain-specific LLMs trained by instruction tuning improve performance in specific fields. For CTI, there is still less comprehensive work to encompass all required parts, including instruction corpora, domain-specific LLMs, and the evaluation benchmark.

In this paper, we introduce a framework to benchmark, elicit, and improve cybersecurity incident analysis and response abilities of **LLM**s for **S**ecurity **Even**ts (called **SEvenLLM**). In Figure 1, the fine-tuned SEvenLLM with fewer parameters can get a more complete and professional response compared to the strong common LLM GPT-3.5, emphasizing the importance of domain-specific LLM for CTI. Specifically, we address the scarcity of high-quality, task-specific datasets for

[1] `https://business.comcast.com/community/`

security event analysis by curating an extensive bilingual (English and Chinese) corpus of cybersecurity incident reports. We first use the LLM (GPT-4) to generate the candidate tasks. Then human experts correct the tasks based on threat intelligence analysis criteria and focal points proposed by security organizations and institutions, such as MITRE[2] and OASIS CTI TC[3], to form a task pool. Given the cybersecurity raw text, we prompt the LLM to select the proper task from the task pool and simultaneously generate the task's query and corresponding response. We refer to this approach as **Select-Instruct**, and we use it to generate the instruction corpus SEvenLLM-Instruct. Open-source LLMs tailored for cyber threat intelligence based on different base models are fine-tuned on the SEvenLLM-Instruct with multi-task learning. To effectively evaluate SEvenLLM for cyber threat intelligence, we construct an evaluation benchmark SEvenLLM-Bench to comprehensively assess the performance of LLMs for CTI, thereby bridging the gap between the common field and the cybersecurity field in understanding and generation.

The main contributions are summarized as follows:

- We create a high-quality bilingual multi-task instructional corpora SEvenLLM-Instruct by crawling cybersecurity texts and adopting the dataset construction method Select-Instruct to address the data scarcity of CTI, where the dataset ensures the practical utility and relevance of the dataset in real-world scenarios.

- Based on the open-source base LLMs, SEvenLLM tailored for cyber threat intelligence is fine-tuned on SEvenLLM-Instruct to assist in the automated and intelligent processing of security incidents. SEvenLLM aims to streamline the analysis process and reduce the reliance on human expertise, thus accelerating and enhancing the analysts' capabilities in threat identification and response.

- To fill in the gaps in the evaluation for cyber threat intelligence, we construct an evaluation benchmark SEvenLLM-Bench comprised of multiple-choice questions and query-answer questions for assessing the performance of LLMs within the context of cybersecurity.

[2]https://attack.mitre.org/
[3]https://oasis-open.github.io/cti-documentation/

2 SEvenLLM

2.1 Model Overview

In Figure 2, we propose a framework (SEvenLLM) for assessing and enhancing the capabilities of LLMs in analyzing and responding to cybersecurity incidents, specifically referred to as security events. First, we collect an extensive bilingual (English and Chinese) dataset of cybersecurity incident reports. Then, we create supervised question-answer pairs. This process involves creating a task pool and using Select-Instruct to generate an instruction corpus SEvenLLM-Instruct. The open-source LLMs (e.g. Llama and Qwen) are further fine-tuned on SEvenLLM-Instruct with multi-task learning objectives tailored for CTI. To accurately assess the effectiveness of SEvenLLM for CTI purposes, we establish a benchmark evaluation, SEvenLLM-Bench, designed to thoroughly evaluate the performance of LLMs in CTI, which aims to close the gap in understanding and generation capabilities between the general and cybersecurity domains.

2.2 Benchmark Construction

Data Collection and Preprocessing To construct the instruction corpora of CTI, we amass a collection of over ten thousand cybersecurity incident websites from leading domestic and international cybersecurity vendors, ensuring the relevance and breadth of the cybersecurity incidents encompassed (from 2004 to 2024). The collection data includes official reports from the websites of security vendors and published media news by Internet enterprises. We design heuristic rules by leveraging language models and threat intelligence features to remove texts that are either too short or too long. After filtering out low-quality records, we deduplicate the collected reports based on their titles and extract textual information from different categories of security event reports. Finally, the curated collection contains $6,706$ English and $1,779$ Chinese high-quality reports as the original corpus.

SEvenLLM-Instruct We are categorizing the analysis of cybersecurity incidents into understanding and generation tasks, encompassing 28 subcategories. The *understanding task* in cybersecurity incident analysis involves transforming unstructured data into structured data by extracting entities and relationships from cybersecurity incidents, aiming

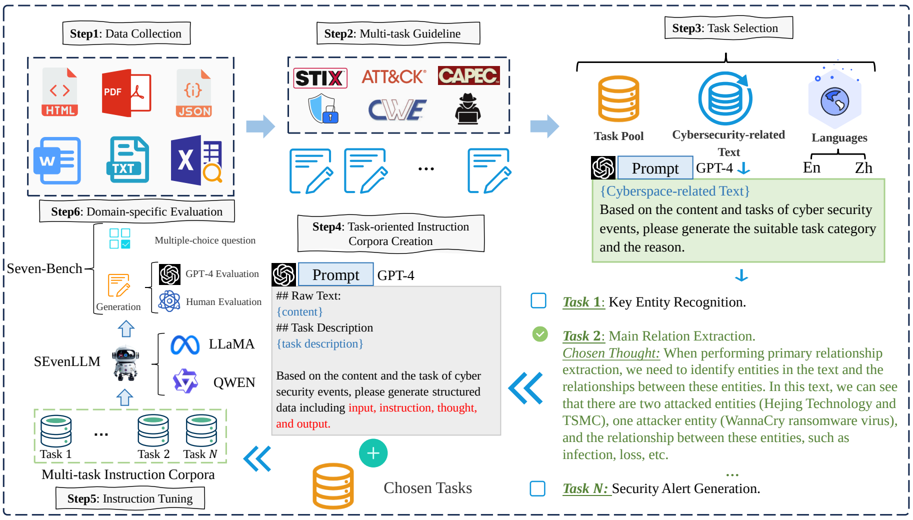


Figure 2: Overview of SEvenLLM. By crawling different formats of files from the Internet, we collect bilingual (English and Chinese) collection of cybersecurity incident reports. First, we adopt LLMs to produce and refine potential tasks to create a task pool. Given raw cybersecurity texts, we use Select-Instruct to select a proper task and generate the query and its answer. We are further fine-tuning the open-source LLMs on SEvenLLM-Instruct with multi-task learning objectives tailored for CTI. A curated CTI evaluation benchmark SEvenLLM-Bench is created to compare SEvenLLM with other baselines.

at acquiring well-defined knowledge such as identifying advanced persistent threat (APT) groups, attack techniques and the relationships between APT groups and techniques. For cybersecurity professionals, the *generation tasks* (e.g., generating summaries and suggesting mitigation measures) provide straightforward knowledge for the general analyst and are used for data fusion and information compression of cybersecurity incident content. We randomly select the raw text feed into the LLM to generate the task name and correct them with human experts. Finally, we design 13 understanding tasks and 15 generation tasks (Detailed definitions of the subcategories can be found in the Appendix A.1). Given the cybersecurity-related raw text, we use the Select-Instruct method to generate the supervised instruction corpora SEvenLLM-Instruct. The sample distribution of the SEvenLLM-Instruct is shown in Figure 3, highlighting the various hierarchical categories and their respective proportions. Dataset examples are provided in Appendix A.2.

SEvenLLM-Bench We use query-answer questions and multiple-choice questions to evaluate the performance of SEvenLLM in CTI. In Table 1, SEvenLLM-Instruct contains nearly 90K samples, and SEvenLLM-Bench has 1300 test samples. Especially for the multiple-choice question (MCQ), 364/1 denotes that the length of the prompt is 364

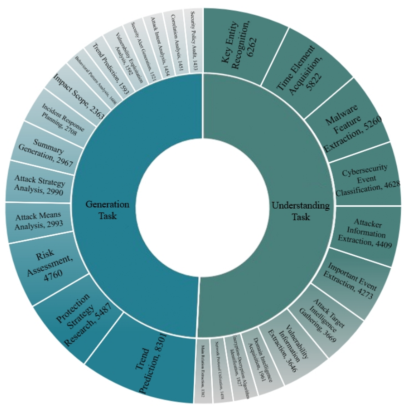


Figure 3: The sample distribution of the SEvenLLM-Instruct.

and the length of the output is 1 (A, B, C, or D). An example of the multiple-choice questions can be found in the Appendix A.3.

Select-Instruct To improve the quality of query-answer datasets, we have refined the self-instruct method specifically for the task of cybersecurity event analysis. Our approach involves two steps. First, we input the raw corpus and use meticulously designed prompts to enable the LLM to select tasks from the task pool, producing task seeds. Second, we input both the task seeds and the raw corpus, prompting the LLM to extract valuable portions of the corpus based on the tasks to generate instructions and answers. While generating instructions and responses, we also require the LLM to provide its reasoning process and justification. (More

	Problem	Zh		En	
		Size	Len	Size	Len
Text	Raw	1779	4798	6706	900
Train	MCQ	3000	347/1	3000	121/1
	QA	41218	384/196	44183	184/90
Test	MCQ	50	364/1	50	138/1
	QA	600	450/236	600	167/91

Table 1: Statistics of datasets. The training data contains two types of questions: multiple-choice questions (MCQ) and query-answer questions(QA). The supervised data is derived from the high-quality original corpus (Text) collected from the Internet. Here, we calculate the length of Chinese at the token level and the length of English by splitting space tokens.

details can be found in Appendix B.1).

2.3 Large Language Model for CTI

Given a cybersecurity raw text $c \in D^r_{all} = \{D^r_{L_i}\}^m_{i=1}$ (L_i is the language of raw text c and D^r_{all} contains Chinese and English in our work), we prompt the LLM $\mathcal{M}$ to select the proper task T_k, where $T_k \in T_{all} = \{T_j\}^n_{j=1}$ and the task pool T_{all} contains 13 understanding tasks and 15 generation tasks. Conditioned on {Pre-defined tasks} and {Cybersecurity-related Raw Text}, we use the prompt "Based on the content and tasks of cyber security events, please generate suitable task category and the reason" in Figure 2 to generate the target task T_k with the chosen reason. Then, we use the raw text c and the chosen task T_k to generate the query, the corresponding response, and the thought q_k, a_k, t_k to construct the supervised fine-tuned corpora $D^s_{all} = \{D^s_i\}^n_{i=1}$, where D^s_{all} is the supervised instruction corpus containing n tasks and $(q_k, a_k) \in D^s_{all}$. In Figure 2, we feed the prompt "{Cybersecurity-related Raw Text}. {Task Definition}. Based on the content and the task of cyber security events, please generate structured data including input, instruction, thought, and output" into the LLM $\mathcal{M}$ to generate the supervised pair q_k, a_k, t_k. The Chain of Thought (CoT) generation process can be described as:

$$P(a_k, t_k|q_k) = P(t_k|q_k) \prod_{j=1}^{J} P(a^j_k|a^{<j}_k; q_k, t_k, \mathcal{M}) \quad (1)$$

where t_k is the reasoning step (Wei et al., 2022; Jin et al., 2024) for the answer a_k. a^j_k is the j-th token of the answer a_k (a_k has J tokens).

2.4 Multitask Instruction Tuning

Given the bilingual multitask instruction corpora $D^s_{all} = \{D^s_i\}^{m,n}_{i=1}$, where D contains $m = 2$ languages and $n = 28$ tasks. The base LLM is jointly trained on the multitask corpus D_{all}:

$$\mathcal{L}_{all} = -\sum_{i=1}^{m} \mathbb{E}_{q^{L_i}_k, a_k, t_k \in D_i} [\log P(a_k, t_k|q_{t_k}; \theta)] \quad (2)$$

where q_k is the query, t_k is the thought, and a_k is the response.

3 Experiment Setting

3.1 Instruction Tuning

Backbone Models Based on the pre-trained model supporting both English and Chinese, we adopt Llama2-7B/13B[9] and Qwen1.5-7B/14B[10] as the foundation model to build SEVENLLM.

Implementation Details To fine-tune all models with different sizes, we set epochs to 3 and batch size to 128. We use a cosine scheduler with a learning rate of 2e-5 and 3% learning rate warmup.

Dataset Based on the Open-source large language model of Llama2 and Qwen1.5, SEVENLLMs are trained on SEVENLLM-Instruct of nearly 90K samples generated by GPT-4.

3.2 Evaluation

F1 Score We use Micro F1 to evaluate the understanding tasks of the LLMs. Let the values from the SEVENLLM-Bench be denoted as the set S_t, and the values generated by the model be denoted as the set S_g. We calculate precision (Precision $= \frac{|S_t \cap S_g|}{|S_g|}$) and recall (Recall $= \frac{|S_t \cap S_g|}{|S_t|}$) separately, and finally compute the Micro F1 score (F1 $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$).

Rouge-L Score We use Rouge-L to evaluate generation tasks of the LLM, which is tokenized by space in English and jieba[11] in Chinese.

Semantic Score Using the multilingual sentence-transformers[12] (Reimers and Gurevych, 2019)

[4] https://huggingface.co/spaces/Qwen/Qwen-Max-0428
[5] https://platform.stepfun.com
[6] https://platform.moonshot.cn
[7] https://www.volcengine.com/product/doubao
[8] https://huggingface.co/google/gemma-2-27b-it
[9] https://github.com/meta-llama/llama
[10] https://github.com/QwenLM/Qwen1.5
[11] https://github.com/fxsjy/jieba
[12] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

Model	Base Model	Model Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
GPT-3.5-Turbo (Schick and Schütze, 2021)	GPT-3.5	🔒	19.2	34.7	26.6	35.1	22.9	34.9	28.9
GPT-4-Turbo (Achiam et al., 2023)	GPT-4	🔒	25.7	28.2	37.4	36.1	31.5	32.1	31.8
GLM-4-Turbo (GLM et al., 2024)	GLM-4	🔒	20.3	30.1	25.8	35.2	23.1	32.7	27.9
Qwen-Max[4]	Qwen-2.5	🔒	18.7	25.3	30.3	33.7	24.5	29.5	27.0
Step-2-16k[5]	Step-2	🔒	22.6	28.4	33.1	35.2	27.8	31.8	29.8
Moonshot-v1-32k[6]	Moonshot-v1	🔒	21.8	30.3	33.3	36.0	27.6	33.2	30.4
Doubao-pro-32k[7]	Doubao-pro	🔒	12.8	27.6	17.6	31.9	15.2	29.8	22.5
Yi-Large-preview(AI et al., 2024)	Yi-large	🔒	22.9	28.1	34.6	36.6	28.7	32.3	30.5
Llama-3.1-Instruct (Dubey et al., 2024)	Llama-3.1	405B	22.6	28.2	34.5	34.1	28.5	31.1	29.8
Hunyuan-MoE-A52B-Instruct(Sun et al., 2024)	Hunyuan-Large	389B	18.7	29.4	30.5	33.4	24.6	31.4	28.0
DeepSeek (DeepSeek-AI, 2024)	DeepSeek-v2	236B	19.2	29.6	33.1	36.5	26.2	33.0	29.6
GLM3 (Zeng et al., 2022)	GLM-3	130B	21.1	31.0	27.3	35.3	24.2	33.2	28.7
Qwen2.5-Instruct (Qwen Team, 2024)	Qwen-2.5	72B	21.4	25.1	32.2	34.9	26.8	30.0	28.4
Llama-3.3-Instruct	Llama-3.3	70B	18.6	29.1	32.5	35.4	25.6	32.2	28.9
Qwen2-57B-A14B-instruct(Yang et al., 2024)	Qwen-2	57B	16.8	31.6	24.3	35.4	20.5	33.5	27.0
Mixtral-8x7B-v0.1(Jiang et al., 2024)	Mixtral-v0.1	56B	19.1	30.2	23.8	19.6	21.5	24.9	23.2
Yi-1.5-Chat-16K	Yi-1.5	34B	18.9	28.0	27.8	32.6	23.3	30.3	26.8
Gemma-2-it[8]	Gemma-2	27B	18.8	27.7	29.3	33.4	24.0	30.5	27.3
Internlm2.5-chat	Internlm-2.5	20B	17.9	29.8	27.5	33.7	22.7	31.8	27.2
Qwen1.5-Chat (Bai et al., 2023)	Qwen-1.5	14B	17.9	30.8	27.9	33.9	22.9	32.3	27.6
Llama2-Chat (Touvron et al., 2023)	Llama-2	13B	8.6	31.0	14.0	28.4	11.3	29.7	20.5
Llama2-Chat	Llama-2	7B	11.3	30.0	14.9	31.4	13.1	30.7	21.9
Qwen1.5-Chat	Qwen-1.5	7B	16.7	30.0	25.7	32.8	21.2	31.4	26.3
SEvenLLM	Llama-2	7B	30.8	37.8	39.5	36.9	35.1	37.3	36.2
SEvenLLM + CoT	Llama-2	7B	29.2	37.5	39.5	37.3	34.3	37.4	35.8
SEvenLLM	Llama-2	13B	**31.1**	37.6	41.4	37.6	**36.2**	37.6	**36.9**
SEvenLLM + CoT	Llama-2	13B	29.7	**38.2**	39.7	**37.7**	34.7	**38.0**	36.3
SEvenLLM	Qwen-1.5	7B	30.4	37.2	40.9	37.2	35.6	37.2	36.4
SEvenLLM	Qwen-1.5	14B	29.7	36.8	**41.6**	37.2	35.6	37.0	36.3

Table 2: Micro F1 & Rouge-L scores(%) of our method and previous baselines for downstream generation tasks.

Model	Base Model	Model Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
GPT-3.5-Turbo	GPT-3.5	🔒	61.6	72.8	69.9	75.0	65.7	73.9	69.8
GPT-4-Turbo	GPT-4	🔒	74.0	**75.1**	82.2	**76.7**	78.1	**75.9**	**77.0**
DeepSeek	DeepSeek-v2	236B	66.8	74.4	77.7	75.7	72.2	75.1	73.6
Qwen1.5-Chat	Qwen	14B	69.5	68.7	76.7	75.9	73.1	72.3	72.7
Llama2-Chat	Llama	13B	48.1	67.9	56.0	70.9	52.0	69.4	60.7
Llama2-Chat	Llama	7B	56.4	67.0	57.7	71.9	57.0	69.4	63.2
Qwen1.5-Chat	Qwen	7B	72.0	69.5	77.1	75.1	74.6	72.3	73.5
SEvenLLM	Llama-2	7B	**76.0**	73.4	81.9	75.9	79.0	74.7	76.9
SEvenLLM + CoT	Llama-2	7B	74.6	73.2	81.3	75.8	77.9	74.5	76.2
SEvenLLM	Llama-2	13B	75.8	73.2	**82.7**	76.0	**79.2**	74.6	76.9
SEvenLLM + CoT	Llama-2	13B	74.4	73.5	81.6	76.0	78.0	74.7	76.4
SEvenLLM	Qwen-1.5	7B	75.8	72.5	81.8	76.2	78.8	74.3	76.6
SEvenLLM	Qwen-1.5	14B	75.2	72.8	82.0	76.0	78.6	74.4	76.5

Table 3: Semantic similarity scores(%) of generation task.

Model	Base Model	Model Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
GPT-3.5-Turbo	GPT-3.5	🔒	81.7	79.9	80.9	76	81.3	77.9	79.6
GPT-4-Turbo	GPT-4	🔒	**84.7**	**87.6**	83.8	**80.5**	84.3	**84.0**	**84.2**
DeepSeek	DeepSeek-v2	236B	84.5	84	**84.6**	79.9	**84.6**	82.0	83.3
Qwen1.5-Chat	Qwen-1.5	14B	82.1	79.6	79.7	77.7	80.9	78.6	79.8
Llama2-Chat	Llama-2	13B	69.2	79.9	69.4	73.3	69.3	76.6	73.0
Llama2-Chat	Llama-2	7B	70.4	78.9	65.2	67.7	67.8	73.3	70.6
Qwen1.5-Chat	Qwen-1.5	7B	78.2	79.5	79.5	77.5	78.9	78.5	78.7
SEvenLLM	Llama-2	7B	82.1	79.7	80.9	77.9	81.5	78.8	80.2
SEvenLLM + CoT	Llama-2	7B	82.3	79.7	79.6	77.5	81.0	78.6	79.8
SEvenLLM	Llama-2	13B	83.7	80.3	80.9	78.0	82.3	79.1	80.7
SEvenLLM + CoT	Llama-2	13B	82.7	80.1	80.3	78.3	81.5	79.2	80.4
SEvenLLM	Qwen-1.5	7B	84.1	80.7	81.3	78.7	82.7	79.7	81.2
SEvenLLM	Qwen-1.5	14B	82.6	79.7	80.6	77.9	81.6	78.8	80.2

Table 4: GPT-4 evaluation of our method and previous baselines downstream generation tasks.

Model	Base Model	Model Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
GPT-3.5-Turbo	GPT-3.5	🔒	22	21	21	21	43	42	85
GPT-4-Turbo	GPT-4	🔒	23	**25**	**25**	**25**	48	**50**	**98**
GLM-4-Turbo	GLM-4	🔒	22	24	24	24	46	48	94
Qwen-Max	Qwen-2.5	🔒	**24**	**25**	24	**25**	48	**50**	**98**
Step-2-16k	Step-2	🔒	23	**25**	23	**25**	46	**50**	96
Moonshot-v1-32k	Moonshot-v1	🔒	23	23	22	24	45	47	92
Doubao-pro-32k	Doubao-pro	🔒	22	24	**25**	**25**	47	49	96
Yi-Large-preview	Yi-large	🔒	**24**	**25**	24	24	48	49	97
Llama-3.1-Instruct	Llama-3.1	405B	23	**25**	24	24	47	49	96
Hunyuan-A52B-Instruct	Hunyuan-Large	389B	23	22	24	22	47	44	91
DeepSeek	DeepSeek-v2	236B	23	24	**25**	**25**	48	49	97
GLM3	GLM-3	130B	23	21	**25**	**25**	**49**	46	94
Qwen2.5-Instruct	Qwen-2.5	72B	23	24	**25**	**25**	48	49	97
Llama-3.3-Instruct	Llama-3.3	70B	**24**	23	22	23	46	46	92
Qwen2-A14B-instruct	Qwen-2	57B	23	22	23	21	46	43	89
Mixtral-8x7B-v0.1	Mixtral-v0.1	56B	**24**	22	22	21	46	43	89
Yi-1.5-Chat-16K	Yi-1.5	34B	23	22	20	23	43	45	88
Gemma-2-it	Gemma-2	27B	23	24	**25**	24	48	48	96
Internlm2.5-chat	Internlm-2.5	20B	21	22	24	20	45	42	87
Qwen1.5-Chat	Qwen	14B	22	24	22	24	44	48	92
Llama2-Chat	Llama	13B	20	21	14	15	34	36	70
Llama2-Chat	Llama	7B	11	18	13	11	24	29	53
Qwen1.5-Chat	Qwen	7B	19	23	**25**	24	44	47	91
SEvenLLM	Llama	7B	22	24	24	**25**	46	49	95
SEvenLLM + CoT	Llama	7B	20	24	23	**25**	43	49	92
SEvenLLM	Llama	13B	23	**25**	23	**25**	46	**50**	96
SEvenLLM + CoT	Llama	13B	22	**25**	24	**25**	46	**50**	96
SEvenLLM	Qwen	7B	23	23	23	24	46	47	93
SEvenLLM	Qwen	14B	22	**25**	24	**25**	46	**50**	96

Table 5: Results of multiple-choice questions.

model (SBERT), we map the text to a 384-dimensional dense vector space. This model can be used for tasks such as clustering or semantic search, allowing for the evaluation of whether the values in a test dictionary can score from a semantic understanding perspective. This approach is more aligned with the open-ended nature of cyber security event analysis. Due to its support for various lengths and both Chinese and English languages, this model is applied to both understanding and generation tasks.

GPT-4 Score To evaluate the answers provided by the baseline model and the trained model, we set the evaluation criteria as {very poor: 1, slightly poor: 2, neutral: 3, good: 4, very good: 5} to assess and compute the score (The detailed instruction is included in Appendix B.2). The final representation result is converted into a percentage format.

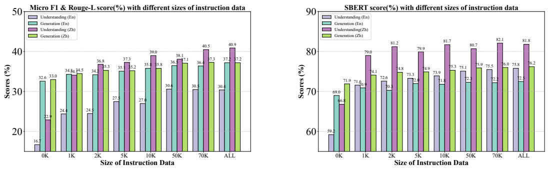
Multiple-choice Question Score We design 100 multiple-choice questions for two languages and two major types of tasks to comprehensively evaluate the model's capabilities. The output results are manually verified to ensure an effective assessment of the model's ability to handle objective questions.

4 Main Results

Micro F1&Rouge-L Evaluation Under rigorous evaluation, SEVENLLM based on Llama2 and Qwen1.5 demonstrates superior performance by fine-tuning on the instruction dataset SEVENLLM-Instruct, as shown in Table 2. This result highlights the strengthening and standardization effects of comprehensive fine-tuning, making the model more suitable for specific tasks in vertical domains. Plus, SEVENLLM + CoT gains no improvement, since the evaluated tasks do not require the complex reason. However, the reasoning text can be provided to cybersecurity staff as a reference. Based on the evaluation results, select a subset of models for semantic evaluation and GPT-4 evaluation, and conduct a further analysis of the outcomes.

Semantic Evaluation Models with larger sizes usually show more robust generalization capabilities. In Table 3, GPT-4 demonstrates a clear advantage in semantic evaluation. Additionally, it is evident that SEVENLLM, after being trained by SEVENLLM-Instruct, still significantly outperforms open-source models and surpasses the performance of both DeepSeek and GPT-3.5.

GPT-4 Evaluation In Table 4, compared to the results of models that have not been trained with SEVENLLM-Instruct, it is evident that training with the datasets provided in this work significantly improves performance. Additionally, among models with less than 20 billion parameters, those



(a) Micro F1 & Rouge-L score. (b) Semantic Similarity score.

Figure 4: Evaluation results of different instruction data sizes.

higher-capacity models can occasionally perform worse in GPT-4 scores. Considering that cybersecurity data is challenging to collect and highly specialized, this underscores the importance of enhancing training materials in this field with datasets like those provided in this paper. It can be observed that GPT-4 scores are relatively close, indicating lower discriminative power, but SEVENLLM still demonstrates superior performance compared to GPT-3.5.

Multiple-choice Question During a revision and comparison of multiple-choice questions, it is observed that SEVENLLM based on Qwen1.5-7B performs less effectively than SEVENLLM based on Llama2-7B, particularly in scenarios requiring a degree of reasoning and where answers may be confusing. The result demonstrates the superior learning capabilities of the Llama2 series models. In Table 5, it is observed that higher-capacity models show improvements in multiple-choice questions. The result indicates that larger-scale and more up-to-date LLMs exhibit superior performance in multiple-choice question tasks. This suggests that enriching overall knowledge and reasoning capabilities with larger model sizes improves performance in multiple-choice questions. It has also been observed that base models can perform correct analysis but still produce incorrect answers, indicating that the full potential of an AI model requires not only general training but also alignment with specialized corpora to yield more accurate outcomes.

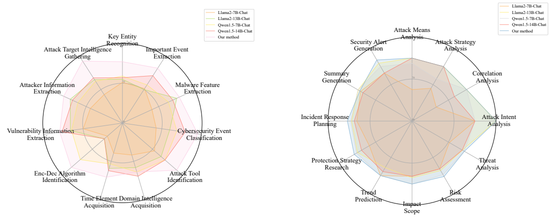
5 Analysis

Training Data Size of Instruction Tuning To discuss the effect of the instruction tuning, we plot Micro F1 & Rouge-L scores and Semantic similarity scores with different training data sizes in Figure 4 (Detailed scores are in Appendix B.5). The size of the instruction data influences the performance. We randomly sample $N = \{0K, 1K, 2K, \ldots, ALL\}$ sentences from the whole corpora to fine-tune the Qwen model. With the increasing training data size, the SEvenLLM performs better. Notably, only $1K$ supervised data sentences significantly improve the zero-shot cross-lingual NER, which benefits from knowledge transfer of the multilingual self-training. When the target annotated corpora size is greater than 10K, our method performs exceptionally well. From the results obtained through Micro F1 & Rouge-L score and Semantic similarity score, the model's performance appears to reach a level close to the testing set when it scales to around the 70K magnitude. Overall, it appears that the larger the training dataset, the better the model's accuracy.

Quality Analysis of SEvenLLM-Bench To ensure the quality of the SEvenLLM-Bench, we employ three experts to evaluate the quality of the test set and calculate the error rate. Manual revision includes corrections for accuracy and reasonableness, inspecting the result content to ensure it meets the requirements of the instructions, is based on prior knowledge from the input, and involves removing redundant information and hallucinations, as well as adding some undetected information. This includes proofreading 100 multiple-choice questions for the correctness and uniqueness of answers. Table 6 describes the distribution of SEvenLLM-Bench, the number of corrections made, and the proportion of these corrections. We can see that the total error rate of SEvenLLM-Bench is 17% generated by GPT-4, and then the experts fix these errors to ensure the correctness of the test set.

Task	|Zh|	|En|	|All|	|Zh (C)|	|En (C)|	|All (C)|	Ratio (%)
Key Entity Recognition	3	25	28	2	1	3	11
Main Relation Extraction	0	0	0	0	0	0	0
Important Event Extraction	12	28	40	2	2	4	10
Malware Feature Extraction	227	41	268	99	9	108	40
Cybersecurity Event Classification	14	86	100	5	3	8	8
Attack Tool Identification	3	21	24	1	2	3	13
Domain Intelligence Acquisition	0	9	9	0	4	4	44
Time Element Acquisition	2	36	38	1	0	1	3
Network Protocol Utilization	0	0	0	0	0	0	0
En-Decryption Algorithm Identification	0	11	11	0	0	0	0
Vulnerability Information Extraction	39	33	72	9	2	11	15
Attacker Information Extraction	20	15	35	12	0	12	34
Attack Target Intelligence Gathering	5	18	23	2	1	3	13
Vulnerability Exploitation Analysis	0	0	0	0	0	0	0
Attack Means Analysis	7	1	8	1	0	1	13
Attack Strategy Analysis	14	3	17	2	0	2	12
Correlation Analysis	7	0	7	0	0	0	0
Attack Intent Analysis	2	1	3	0	0	0	0
Threat Analysis	224	92	316	27	10	37	12
Risk Assessment	10	37	47	3	1	4	9
Impact Scope	8	20	28	0	2	2	7
Trend Prediction	2	9	11	1	1	2	18
Behavioral Pattern Analysis	0	0	0	0	0	0	0
Protection Strategy Research	39	50	89	0	3	3	3
Incident Response Planning	6	46	52	0	2	2	4
Security Policy Audit	0	1	1	0	0	0	0
Summary Generation	6	55	61	2	13	15	25
Security Alert Generation	0	12	12	0	2	2	17
Total	650	650	1300	168	54	222	17

Table 6: Correction of SEvenLLM-Bench. |Zh| denotes the number of Chinese samples in SEvenLLM-Bench while $|\text{Zh}(C)|$ denotes the number of samples need to be corrected.



(a) Understanding tasks.   (b) Generation tasks.

Figure 5: Comparison between SEvenLLM with other models.

Human Evaluation To better evaluate the LLM Llama2-Chat, Qwen1.5-Chat and SEvenLLM(based on Llama2-7B), we employ five volunteers to score the test set SEvenLLM-Bench of cyber threat intelligence {1: very poor; 2: poor; 3: neutral, 4: good; 5: very good} based on three criteria (1) Correctness: Correctness refers to the accuracy and reliability of the information provided or the actions performed. (2) Fluency: It's about how smoothly and coherently the system can produce or interpret language, making it comprehensible and pleasant for human users. (3) Instruction Following Capability: This component assesses how effectively a system or application can comprehend and execute commands or requests given by users. Based on the results of manual evaluations, we have compiled the performance statistics of Llama2-Chat, Qwen1.5-Chat, and SEvenLLM across various sub-tasks, as illustrated in Figure 5. The results indicate a significant improvement in SEvenLLM across most tasks, particularly in specialized domains. However, it is also observed that in general-purpose tasks, such as **Domain Intelligence Acquisition** and **Summary Generation**, the open-source large language models demonstrate slight advantages (See Appendix B.4 for details).

Additionally, in Table 7, we calculate the correlations between the scores assigned by GPT-4 and human evaluators for each question using Spearman and Pearson coefficients. It is evident that the Understanding Task, which has more definitive answers, shows a stronger correlation, whereas the Generation Task exhibits greater subjectivity. Overall, a strong and statistically significant correlation exists between **GPT-4 Score** and **Human Evaluation**, suggesting that GPT-4's scoring is consistent with human judgment.

Pairwise Comparison To enhance GPT-4's discriminative ability and evaluate SEvenLLM

Model	Understanding Task		Generation Task		Average	
	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
SEvenLLM	0.7094	0.7288	0.5780	0.6434	0.6534	0.6999
Llama2-7B-Chat	0.8299	0.9177	0.6024	0.6813	0.7445	0.8591
Llama2-13B-Chat	0.5721	0.4559	0.6508	0.6524	0.5820	0.4677
Qwen1.5-7B-Chat	0.6500	0.7048	0.5070	0.5941	0.5902	0.6489
Qwen1.5-14B-Chat	0.6000	0.6279	0.5099	0.4680	0.5666	0.5724

Table 7: Spearman and Pearson Correlations between GPT-4 Scores and Human Evaluation. All measures in this table are statistically significant to $p < 0.01$.

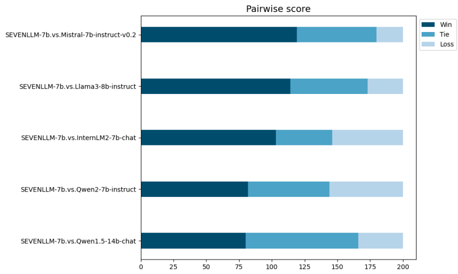


Figure 6: Pairwise scores of different models on Q&A test in SEvenLLM-Bench.

(based on Qwen1.5-7B) against newly released open-source LLMs, we select 200 Q&A samples from the SEvenLLM-Bench for pairwise comparison. We choose four recently released LLMs of comparable size and strong capabilities: Qwen2-7b-instruct[13], InternLM2-7b-chat[14], Llama3-8b-instruct[15], and Mistral-7b-instruct-v0.2[16]. We include Qwen1.5-14B-Chat, which scored highly according to GPT-4 In Table 4. For the pairwise competition method, first, we select two models to be assessed and have them generate answers based on the given questions. Then, we use GPT-4 as the scoring model to judge which of the two assessed models provides the better answer (See Appendix B.3 for details). As shown in Figure 6, SEvenLLM outperforms the other well-performing models, demonstrating the current superiority of our approach.

6 Related Work

Cyber Threat Intelligence Cyber threat intelligence (Liu and Zhan, 2023; Motlagh et al., 2024; Siracusano et al., 2023; Camacho et al., 2024) has emerged as a pivotal aspect of cybersecurity practices aimed at understanding and anticipating cyber threats for proactive defense. This field encompasses collecting, analyzing, and disseminating information regarding current or potential attacks threatening the security of information systems (Park et al., 2023; Guo et al., 2021). By leveraging CTI, organizations are equipped to identify specific threats and vulnerabilities, enabling them to tailor their security measures more effectively. This intelligence-gathering process draws from a wide array of sources, including historical cyber attack data, hacker forums, malware samples, and more (Joyce et al., 2023), to provide a comprehensive view of the cyber threat landscape, which helps in enhancing the security posture against known threats and aids in predicting and mitigating future cyber attacks.

Domain-specific Large Language Model The development of the generative pre-trained Transformer (GPT) series by OpenAI progressively expands the frontiers of the field of natural language processing (NLP), such as named entity recognition (Mo et al., 2024), machine translation (Yang et al., 2021) and text style transfer (Wang et al., 2020). It even has an impact (Zhai et al., 2024) on some traditional recommendation and advertising methods (Wang et al., 2023b). To further enhance the adaptability of LLMs to various tasks, instruction-tuning has become a commonly adopted method by utilizing instructions derived from human-curated prompts, feedback, and public benchmarks, which supports the fast development of the domain-specific LLMs, such as coding (Roziere et al., 2023), IT operation (Guo et al., 2023), and science (Xie et al., 2023). Instruction tuning (Wang et al., 2023a) is introduced to generate novel tasks and associated instructions from the ground up, offering performance and cost-efficiency advantages.

7 Conclusion

In this work, we present SEvenLLM, which marks an advancement in the use of LLMs for benchmarking, eliciting, and improving cybersecurity incident analysis and response abilities in cybersecurity LLMs. By leveraging a meticulously curated bilingual instruction corpus from an extensive collection of cybersecurity texts, SEvenLLM-Instruct bridges the gap in the availability of effective data for cybersecurity applications. Extensive experiments on a specialized cybersecurity benchmark, SEvenLLM-Bench, corroborate the efficacy of SEvenLLM in improving analytical capabilities and providing robust responses to cyber threats.

[13]https://huggingface.co/Qwen/Qwen2-7B-Instruct
[14]https://huggingface.co/internlm/internlm2-7b
[15]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[16]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Limitations

The primary data source collected focuses on English as the mainstream language for cybersecurity intelligence. Although it has been extended to include Chinese data, the analysis of cybersecurity incidents would benefit from a multilingual capability, which requires further collection and organization. In the future, we will expand SEvenLLM-Instruct to more languages.

Ethical Considerations

In this study, we analyze cyber threat intelligence exclusively derived from publicly available sources provided by security companies. Although some sensitive cybersecurity terminology is used, we strictly avoid involving any personal privacy, trade secrets, or activities that can constitute a threat. Moreover, we ensure that all data handling and usage comply with ethical standards and legal regulations to maintain transparency and integrity in our research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Shahid Alam. 2022. Cybersecurity: Past, present and future. *arXiv preprint arXiv:2207.01227*.

Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3971–3988. USENIX Association.

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*.

Jose Manuel Camacho, Aitor Couce-Vieira, David Arroyo, and David Rios Insua. 2024. A cybersecurity risk analysis framework for systems with artificial intelligence components. *arXiv preprint arXiv:2401.01630*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools.

Hongcheng Guo, Xingyu Lin, Jian Yang, Yi Zhuang, Jiaqi Bai, Tieqiao Zheng, Bo Zhang, and Zhoujun Li. 2021. Translog: A unified transformer-based framework for log anomaly detection. *arXiv preprint arXiv:2201.00016*.

Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, et al. 2023. Owl: A large language model for it operations. *arXiv preprint arXiv:2309.09298*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Feihu Jin, Yifan Liu, and Ying Tan. 2024. Zero-shot chain-of-thought reasoning guided by evolutionary algorithms in large language models. *arXiv preprint arXiv:2402.05376*.

Robert J. Joyce, Dev Amlani, Charles Nicholas, and Edward Raff. 2023. MOTIF: A malware reference dataset with ground truth family labels. *Comput. Secur.*, 124:102921.

Jiehui Liu and Jieyu Zhan. 2023. Constructing knowledge graph from cyber threat intelligence using large language model. In *IEEE International Conference on Big Data, BigData 2023, Sorrento, Italy, December 15-18, 2023*, pages 516–521. IEEE.

Ying Mo, Jian Yang, Jiahao Liu, Qifan Wang, Ruoyu Chen, Jingang Wang, and Zhoujun Li. 2024. MCL-NER: cross-lingual named entity recognition via multi-view contrastive learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18789–18797.

Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. 2024. Large language models in cybersecurity: State-of-the-art. *arXiv preprint arXiv:2402.00891*.

Kuchan Park, Junho Hong, Wencong Su, and HyoJong Lee. 2023. Machine learning based post event analysis for cybersecurity of cyber-physical system. *arXiv preprint arXiv:2311.13488*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *NAACL 2021*, pages 2339–2352.

Giuseppe Siracusano, Davide Sanvito, Roberto Gonzalez, Manikantan Srinivasan, Sivakaman Kamatchi, Wataru Takahashi, Masaru Kawakita, Takahiro Kakumaru, and Roberto Bifulco. 2023. Time for action: Automated analysis of cyber threat intelligence in the wild. *arXiv preprint arXiv:2307.10214*.

Xingwu Sun, Yanfeng Chen, Yiqing Huang, Ruobing Xie, Jiaqi Zhu, Kai Zhang, Shuaipeng Li, Zhen Yang, Jonny Han, Xiaobo Shu, et al. 2024. Hunyuan-large: An open-source moe model with 52 billion activated parameters by tencent. *arXiv preprint arXiv:2411.02265*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508.

Yunli Wang, Zhiqiang Wang, Jian Yang, Shiyang Wen, Dongying Kong, Han Li, and Kun Gai. 2023b. Adaptive neural ranking framework: Toward maximized business goal for cascade ranking systems. *arXiv preprint arXiv:2310.10462*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2020. Formality style transfer with shared latent space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249.

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

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. 2023. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 446–455.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*.

Jun Zhao, Qiben Yan, Xudong Liu, Bo Li, and Guangsheng Zuo. 2020. Cyber threat intelligence modeling based on heterogeneous graph convolutional network. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2020, San Sebastian, Spain, October 14-15, 2020*, pages 241–256. USENIX Association.

A Datasets

A.1 Task list

The instruction dataset tasks are categorized into 2 main types and 28 subtasks. The names and definitions of these tasks are described in Table 8.

Task Name	Task Description
	Understanding Task
(1) Key Entity Recognition	Identify the main entity information in the text, such as attacker organization, victim type, main person, the common vulnerabilities and exposures (CVE), email address, virtual account, IP address, and the indicators of compromise (IOCs).
(2) Main Relation Extraction	Extract the relationships between major entities such as attacker, victim, attack method and so on. Through relationship extraction, connections between entities can be established to help cybersecurity experts better understand the content and context of threat intelligence.
(3) Important Event Extraction	Key information such as the type, time, location, and impact of the event can be identified through critical event extraction.
(4) Malware Feature Extraction	Extract the malware features mentioned in the text, such as file characteristics, means of exploitation, and harm caused.
(5) Cybersecurity Event Classification	The classification results or category characteristics such as event type, severity, etc. are extracted from the security event information and structurally labeled.
(6) Attack Tool Identification	Tools and toolchains utilized in the attack are identified and extracted.
(7) Domain Intelligence Acquisition	Domain names often involve information about phishing sites and locations, obtaining the domain name used by the attacker to look for potential relevance.
(8) Time Element Acquisition	The timing of cybersecurity events is extracted and the timeline is scrutinized and parsed in detail.
(9) Network Protocol Utilization	Extract the network protocols used in the attack, which may include some information containing the attacker's characteristics.
(10) Enc-Dec Algorithm Identification	The process of identifying the encryption or decryption algorithm used in a text and determining the specific algorithm used for the encryption or decryption operation in the text.
(11) Vulnerability Intelligence Extraction	Identify the main information of the vulnerability described in the text, the vulnerability number, and the way the vulnerability is exploited.
(12) Attacker Information Extraction	Characterize the attacker's place of origin, ip, language, unique fingerprints during the attack, and attack behavior and confrontation ideas
(13) Attack Target Intelligence Gathering	Analyze the main characteristics of the attacked target, such as operating system, software, and target industry, field, geographical location, etc.
	Generation Task
(1) Vulnerability Exploitation Analysis	Analyze and assess the exploitation of known vulnerabilities (also known as exploits).
(2) Attack Means Analysis	Analyze the means and specific methods used in attacks during cybersecurity incidents.
(3) Attack Strategy Analysis	Analyze the attacker's tactics, attack plan, or usual methods in a cybersecurity incident.
(4) Correlation Analysis	Analyze the connections and correlated evidence between different threat intelligence reports and cybersecurity incidents.
(5) Attack Intent Analysis	Analyze the attacker's potential motivation, intent, target industry, or target area.
(6) Threat Analysis	Analyze potential threats and possible hazards in cybersecurity incidents.
(7) Risk Assessment	Assess the risk and level of risk posed by the security incident or such attack.
(8) Impact Scope	Analyze the scope and impact of security incidents.
(9) Trend Prediction	Predicts future security trends based on historical and current data, or predicts likely development outcomes based on known information.
(10) Behavioral Pattern Analysis	Research the behavioral pattern characteristics of attackers in cyber security incidents, including attack timing, target selection, etc.
(11) Protection Strategy Research	Research and design strategies to protect against such security incidents or attacks.
(12) Incident Response Planning	Develop an emergency response plan to deal with such incidents.
(13) Security Policy Audit	Audit and evaluate the applicability and effectiveness of proposed security policies.
(14) Summary Generation	A concise and complete summary of the text needs to be generated based on the content of the long text, highlighting the activities and characteristics associated with the main cybersecurity incidents.
(15) Security Alert Generation	Automatically generate security alerts based on analysis of events.

Table 8: List of the understanding and generation tasks.

A.2 SEvenLLM-Instruct

This section presents examples of the Understanding Task and the Generation Task respectively.

```
# Understanding Task

{
    "category": "Malware Feature Extraction",
    "instruction": "Execute the following information extraction task
        and return the results in json format: Identify and extract
       characteristics of malware, their delivery methods, exploited
       vulnerabilities, and the potential damage caused.",
    "input": "One of the most ingenious delivery methods for malware
       today is by exploit kit. Exploit kits are sneaky little
       suckers that rummage around in your computer and look for
       weaknesses in the system, whether that's an unprotected
       operating system, a software program that hasn't been updated
       in months, or a browser whose security protocols aren't up to
       snuff. Malware like ransomware can cause irreversible damage
       by encrypting files without a backup. Malvertising, or
```

```
        malicious ads, is another method that can deliver exploit kits
         to your computer without needing a click. Social engineering,
         such as phishing emails or tech support scams, can also lead
        to malware infections.",
    "thought": "The task of malware feature extraction requires
        identifying specific types of malware mentioned, their methods
         of delivery, the vulnerabilities they target, and the type of
         damage they inflict. This involves scanning the text for
        relevant terms and phrases that describe malware features and
        categorizing them accordingly.",
    "output": {
        "Malware Types": [
            "Exploit Kits",
            "Ransomware"
        ],
        "Delivery Methods": [
            "Malvertising",
            "Phishing Emails",
            "Tech Support Scams"
        ],
        "Exploited Vulnerabilities": [
            "Unprotected Operating System",
            "Outdated Software",
            "Weak Browser Security Protocols"
        ],
        "Potential Damage": [
            "Encryption of Files without Backup",
            "Unauthorized System Access"
        ]
    }
}
```


```
# Generation Task
```


```
{
    "category": "Threat Analysis",
    "instruction": "Execute the following analysis generation task
        and return the result in text format: Analyze the
        cybersecurity incident involving the Flash EK exploit kit and
        describe its threat mechanism and potential impact.",
    "input": "A new exploitation pattern was identified where the
        payload was distributed without a typical infection chain,
        bypassing the usual landing page. This activity dated back to
        at least September 15 and was confirmed to involve Flash EK. A
         tool named Sulo, used for dynamic Flash file analysis,
        revealed that the payload was Flash-based, minimizing its
        footprint and complicating traceability. The exploit kit had
        recently dropped Java exploits, making it stealthier.
        Malwarebytes Anti-Exploit can block the SWF exploit. Two
        threads exist in Flash EK: a classic one and a new entirely
        Flash-based one.",
```

```
    "thought": "When analyzing this incident, I should focus on how
       the Flash EK operates differently from traditional exploit
       kits by using a Flash-based payload delivery system. I need to
        consider the implications of such a method, including its
       minimal footprint and the challenges it poses for detection
       and analysis. Additionally, the potential impact of this
       stealthier approach on users and the effectiveness of existing
        security measures like Malwarebytes Anti-Exploit should be
       assessed.",
    "output": "The Flash EK exploit kit represents a sophisticated
       threat with its new exploitation pattern that deploys payloads
        directly through Flash files, circumventing conventional
       infection chains. This approach reduces its digital footprint
       and makes forensic analysis more challenging. The use of a
       Flash-based mechanism also indicates an adaptation strategy by
        the threat actors to evade detection and enhance the exploit'
       s effectiveness. The recent abandonment of Java exploits by
       Flash EK suggests a shift towards more stealthy operations.
       Security solutions like Malwarebytes Anti-Exploit are capable
       of blocking the SWF exploit, but the existence of two threads
       within Flash EK, including the newer Flash-only variant,
       requires continuous vigilance and updates to defensive
       measures."
}
```


A.3 Multiple-choice question

This section presents an example of the multiple-choice questions.


```
{
    "category": "Important Event Extraction",
    "instruction": {
        "question": "What is the major development concerning
           Microsoft Threat Protection mentioned in the text? Please
           choose a correct answer from A, B, C, D.",
        "choice": {
            "A": "Integration with new email service platforms",
            "B": "Deployment of a new antivirus software",
            "C": "Enhancement of the Azure Security Center",
            "D": "Partnership with a new cybersecurity firm"
        }
    },
    "input": "December was another month of significant development
       for Microsoft Threat Protection capabilities. As a quick recap
       , Microsoft Threat Protection is an integrated solution
       securing the modern workplace across identities, endpoints,
       user data, cloud apps, and infrastructure. Last month, we
       shared updates on capabilities for securing identities,
       endpoints, user data, and cloud apps. This month, we provide
       an update for Azure Security Center which secures
       organizations from threats across hybrid cloud workloads.
       Additionally, we overview a real-world scenario showcasing
       Microsoft Threat Protection in action.",
```

```
    "thought": "The input specifically mentions an update provided
       for Azure Security Center to secure organizations from threats
        across hybrid cloud workloads. This represents a major
       development concerning Microsoft Threat Protection
       capabilities and is thus the major event to be extracted from
       the text.",
    "output": "C"
}
```

A.4 Human Annotation

We engaged five experts in the field of cybersecurity to annotate SEvenLLM-Bench and correct the data generated by GPT-4 from our laboratory. We compensated them $3 per data entry and provided GPT-4 interface for the annotation tasks. In total, we annotated approximately 1,300 data entries, with a total cost of around $4,000. The annotators were also responsible for checking the quality of the questions and answers, including tasks such as defining cybersecurity event analysis, generating instructions, and corresponding responses. We clarified the source of the data and its legal usage to all participants, and required the annotators to maintain confidentiality until the article's publication.

B Instructions

B.1 Select-Instruct

The task seed generation prompt we designed is listed below. Using our method, out of 44,240 generated data points, there were only 24 erroneous data points and 37 empty outputs. In contrast, the self-instruct method, when used to generate instructions and answers for all tasks, produced 41,296 data points, of which 78 are erroneous and 1,182 are empty outputs.

[Instruction]
Based on the given content, combined with your knowledge of cyberspace security, select four extraction tasks and three generation tasks that best fit the content of cybersecurity incidents. The task list is as follows:
Understanding Tasklist:
[
  "Key Entity Recognition",
  "Main Relation Extraction",
  ...
  "Attack Target Intelligence Gathering"
]
Generation Tasklist:
[
  "Vulnerability Exploitation Analysis",
  "Attack Means Analysis",
  ...
  "Security Alert Generation"
]

[Example]
{
  "Understanding Task": [
    {
      "category": "Key Entity Recognition",

"thought": "The task of key entity recognition in cyber security incidents needs to identify information such as attack organizations, related software, main characters, virtual accounts, emails, etc. in the text"
}],
"Generation Task": [
{
"category": "Summary Generation",
"thought": "This text mainly introduces that Anheng Threat Intelligence Center discovered and reported a new 0Day vulnerability, which affects multiple versions of Windows 10, including the latest 20H2 fully patched version. When generating the summary, I needed to highlight the novelty of the vulnerability, its scope of impact, and the interaction between the Arnhem Threat Intelligence Center and Microsoft."
}]
}

[Notice]
Please generate a suitable task based on the text content and provide a chain of thought for selecting this task.
The generated results are expressed in the form of a dictionary combining list and json just as the sample.

[Input]
replace with your input here

[Answer]

After obtaining the sub-tasks corresponding to each cybersecurity incident, a combination of one-shot and chain-of-thought approaches is employed to generate a Q&A dataset based on the context. The constructed prompts are illustrated as follows.

[Instruction]
Based on the given content and the task described next, combined with your knowledge of cyberspace security, focusing on high-quality instructions and output results.

[Example]
{
"category": "Key Entity Recognition",
"instruction": "Execute the following information extraction task and return the results in json format: Analyze and extract key entities in cyber security incidents.",
"input": "The ATW organization was established in June 2021...and steal related source code, data, etc. The relevant information can be used to further exploit and penetrate the involved network information system, which is a typical supply chain attack.",
"thought": "The task of key entity recognition in cyber security incidents needs to identify information such as attack organizations, related software, main characters, virtual accounts, emails, etc. in the text",
"output": {
"Attack Organization": "ATW Organization",
"Related Software": ["SonarQube","Gogs","Gitblit"],

"Email":["AgainstTheWest@riseup.net","apt49@riseup.net"],
"Main Characters":["Tillie Kottmann","Pawel Duda"],
"URL":["https://t.me/s/ATW2022"],
"Virtual Accounts": ["AgainstTheWest"],
"Attacker Regions": ["Switzerland","France","Poland","Canada"]
}
}

[Explanation]
* category: Indicates the task type. Use the given task category.
* instruction: It is an instruction generated for this task and is required to be as common as possible in all network security incident analysis problems.
* input: It is an excerpt of valuable information from the text. It must be a complete paragraph and contain as much information as possible.
* thought: It is a chain of thinking that involves step-by-step thinking in the process of generating output based on input based on the understanding of task category and instruction.You can use the given thought directly, or optimize based on instruction and input.
* output: It requires you to generate an appropriate form of answer based on your thought chain according to the requirements of the instruction and the content in the input. The output result must be consistent with the json format in the text.

[Notice]
The output of this sample is only for the Key Entity Recognition task to clarify your overall output format. For the sample results of other tasks, please ensure that the result form is correct according to the task type. For example, when extracting relationships, it is best to use relational triples to represent the output. Each generated sample is represented by a json and can be placed in one line.

[Input]
The given content is: {replace with your input here}
The category of task that need to be performed based on a cybersecurity incident is: {category}.
Regarding this task and the content of network security incidents, the thought you can refer to is: {thought}.

[Output]

B.2 GPT-4 Score

We design a detailed scoring criterion for GPT-4 to evaluate the results generated by SEvenLLM. To ensure that the scoring of GPT-4 is as objective and effective as possible, the answers must be modified when not awarded full marks.

[Instruction]
As a knowledge analysis expert in the field of cybersecurity, please rate the following network security event Q&A results. The focus is on evaluating whether the provided answers align with the questions, whether the content is accurate, readable and valuable, and provide reasons. The required scoring range is from 1 to 5 points. If full marks are not given, optimized results must be given. Returns a json format result.
Scoring method:

Using a 5-point scoring system, 1 point: very poor; 2 points: slightly poor; 3 points: usable; 4 points: good; 5 points: very good
Scoring reference criteria:
* Whether the answer is answered accurately as required by the question, whether the format is regular, and whether the language is coherent and readable.
* Whether the problem belongs to the field of network security incident analysis, whether the description is clear and has certain analytical significance.
* Whether the overall content has information value in the field of network security and whether it is challenging and difficult.

[Example]

```
{
    "score": "",
    "reason": "",
    "improved result":""
}
```

[Explanation]
* **score**: Give a fraction, expressed numerically.
* **reason**: Give the reason for the score. If it fails to get a full score or is worse, please give a deduction point and describe the optimization results.
* **Improved output**: Give a better result based on the reason description to solve the shortcomings you mentioned. The format is required to be consistent with the output. When the score is lower than 5 points, improved output must be provided.

[Notice]
The modified data must be complete. When no modification is needed, it can be represented by an empty string. A complete json format result must be generated.

[Input]
The content of the network security incident is: {input}
The question raised regarding the content of this cybersecurity incident is: {instruction}
The answer provided for this event's content and question is: {output}

[Answer]

B.3 Pairwise Score

As our work progressed, more powerful open-source LLMs have been released. To verify that SEvenLLM maintains its advantages among models of similar scale, we select 200 test samples from the SEvenLLM-Bench, comprising 50 tasks each of Chinese understanding, English understanding, Chinese generation, and English generation. GPT-4 is tasked with selecting the best answer from those generated by SEvenLLMand the comparative models. The constructed prompts are shown below, considering the question content, the answers from the two models, and the reference answer. We then tally the number of victories, ties, and losses for SEvenLLM.

[Instruction]
Please act as an impartial judge and evaluate the quality of the responses provided by two AI

assistants to the user question displayed below. Your evaluation should consider the accuracy of the content, the conformity to formatting standards, and the consistency of the language used. You will be given a reference answer, assistant A's answer, and assistant B's answer. Your job is to evaluate which assistant's answer is better. Begin your evaluation by comparing both assistants' answers with the reference answer. Identify and correct any mistakes. Whether the answer format complies with the requirements of the instructions is also considered. Avoid any positional biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. You can provide your explanation in 'reason'. After providing your explanation, output your final verdict by strictly following this format: [[A]]: assistant A is better, [[B]]: assistant B is better, and [[C]]: Two answers tie.

[Input]
[User Question]
Below is an instruction that describes a task, paired with an input that provides further context.
Write a response that appropriately completes the request.
Instruction:
{instruction}
Input:
{input}
[Assistants Output]
[The Start of Reference Answer]{Reference_answer}[The End of Reference Answer]
[The Start of Assistant A's Answer]{Answer_A}[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]{Answer_B}[The End of Assistant B's Answer]

Please note that you are evaluating which answer is better based on the question, not answering the [User Question]. Your answer is '[[A]]', '[[B]]' or '[[C]]' with json format.
The answer example is :
{
"answer": "[[A]]",
"reason":""
}

[Answer]

The detailed results of the pairwise comparison are provided in Table 9. The Win, Tie, and Loss in the table represent the numbers obtained by SEVENLLM.

Compared Models	Win	Tie	Loss
SEVENLLM-7b.vs.Qwen1.5-14b-chat	80	86	34
SEVENLLM-7b.vs.Qwen2-7b-instruct	82	62	56
SEVENLLM-7b.vs.InternLM2-7b-chat	103	43	54
SEVENLLM-7b.vs.Llama3-8b-instruct	114	59	27
SEVENLLM-7b.vs.Mistral-7b-instruct-v0.2	119	61	20

Table 9: The results of pairwise comparison.

B.4 Human Evaluation

The detailed evaluation results by human experts are presented in Table 10.

Task Category	Task	Llama2-7B-Chat	Llama2-13B-Chat	Qwen1.5-7B-Chat	Qwen1.5-14B-Chat	SEVENLLM-Qwen-7B
Understanding Task	**Key Entity Recognition**	70.4	72.0	73.6	72.8	80.8
	Important Event Extraction	70.6	70.0	72.8	78.3	82.8
	Malware Feature Extraction	66.3	80.0	75.4	77.8	81.4
	Cybersecurity Event Classification	67.1	74.0	77.6	81.8	88.2
	Attack Tool Identification	71.3	75.7	78.3	78.3	86.1
	Domain Intelligence Acquisition	66.7	73.3	75.6	77.8	75.6
	Time Element Acquisition	65.7	73.7	71.4	74.9	78.3
	Enc-Dec Algorithm Identification	60.0	62.0	78.0	62.0	84.0
	Vulnerability Information Extraction	69.9	76.4	75.8	80.6	81.8
	Attacker Information Extraction	67.7	78.8	74.6	77.6	82.4
	Attack Target Intelligence Gathering	67.0	76.0	75.0	77.0	87.0
Generation Task	**Attack Means Analysis**	65.0	80.0	77.5	80.0	80.0
	Attack Strategy Analysis	68.0	80.0	73.3	80.0	80.0
	Correlation Analysis	63.3	80.0	80.0	73.3	80.0
	Attack Intent Analysis	80.0	90.0	80.0	80.0	90.0
	Threat Analysis	70.5	77.7	78.7	77.6	78.8
	Risk Assessment	76.3	77.2	79.5	76.3	80.5
	Impact Scope	76.3	77.0	80.0	76.3	80.0
	Trend Prediction	80.0	75.6	80.0	77.8	80.0
	Protection Strategy Research	73.5	80.0	80.5	77.8	81.8
	Incident Response Planning	76.3	78.8	79.2	77.5	80.8
	Summary Generation	78.3	79.7	79.7	77.0	78.7
	Security Alert Generation	76.4	81.8	80.0	76.4	83.6

Table 10: Human Evaluation of different models on understanding and generation tasks.

B.5 Training Data Size of Instruction Tuning

The scores of SEVENLLM across different scales of training data are presented in Table 11 and Table 12.

Base Model	Training Data Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
Qwen1.5-7B-base	0	16.7	32.6	22.9	33.0	19.8	32.8	26.3
Qwen1.5-7B-base	1,000	24.4	34.3	34.1	34.5	29.2	34.4	31.8
Qwen1.5-7B-base	2,000	24.5	34.2	36.8	35.3	30.6	34.8	32.7
Qwen1.5-7B-base	5,000	27.5	35.1	37.3	35.2	32.4	35.2	33.8
Qwen1.5-7B-base	10,000	27.0	35.8	39.0	35.8	33.0	35.8	34.4
Qwen1.5-7B-base	50,000	30.6	36.5	38.1	37.1	34.4	36.8	35.6
Qwen1.5-7B-base	70,000	30.5	36.4	40.5	37.3	35.5	36.9	36.2
Qwen1.5-7B-base	All	30.4	37.2	40.9	37.2	35.6	37.2	36.4

Table 11: Micro F1 & Rouge-L scores(%) of models across different training data sizes.

Base Model	Training Data Size	Understanding (En)	Generation (En)	Understanding (Zh)	Generation (Zh)	Understanding (Avg.)	Generation (Avg.)	Avg.
Qwen1.5-7B-base	0	59.2	69.0	66.8	71.9	63.0	70.4	66.7
Qwen1.5-7B-base	1,000	71.6	70.9	79.0	74.1	75.3	72.5	73.9
Qwen1.5-7B-base	2,000	72.6	70.3	81.2	74.8	76.9	72.6	74.7
Qwen1.5-7B-base	5,000	73.3	71.9	79.9	74.9	76.6	73.5	75.0
Qwen1.5-7B-base	10,000	73.9	71.8	81.7	75.3	77.8	73.5	75.7
Qwen1.5-7B-base	50,000	75.1	72.3	80.7	75.9	77.9	74.1	75.9
Qwen1.5-7B-base	70,000	75.5	72.2	82.1	76.0	78.8	74.1	76.4
Qwen1.5-7B-base	All	75.8	72.5	81.8	76.2	78.8	74.3	76.6

Table 12: Semantic similarity scores(%) of models across different data sizes.

C Raw corpus

Our cybersecurity raw corpus is provided by security vendors and consists of open-source security event reports from global security companies. These reports may contain sensitive terms, such as names of malware. We ensure that these contents do not include any private information and are harmless. Some example source websites are listed below:

- https://cybersecurity.att.com/blogs/labs-research/
- https://www.cyfirma.com/research/
- https://securelist.com/tag/apt/
- https://www.prodaft.com/resources/latest-reports
- https://asec.ahnlab.com/en/feed/
- https://www.intrinsec.com/category/cyber-threat-intelligence
- https://www.fortinet.com/content/fortinet-blog/us/en/threat-research
- https://research.checkpoint.com/category/threat-research-2/
- https://www.microsoft.com/en-us/security/blog/
- https://www.proofpoint.com/us/blog/threat-insight
- https://www.venustech.com.cn/new_type/gjywxfx/
- https://www.antiy.com/response.html
- https://www.zscaler.com/blogs/security-research
- https://www.mcafee.com/blogs/other-blogs/mcafee-labs/page/1/
- https://www.trendmicro.com/en_us/research.tagSearch.json