An Empirical Study of Many-to-Many Summarization with Large Language Models

Jiaan Wang[1], Fandong Meng[1]*, Zengkui Sun[4], Yunlong Liang[1], Yuxuan Cao[5]
Jiarong Xu[2], Haoxiang Shi[3] and Jie Zhou[1]
[1]Pattern Recognition Center, WeChat AI, Tencent Inc, China [2]Fudan Unversity
[3]Inner Mongolia University of Technology [4]Beijing Jiaotong University [5]Zhejiang University
{torchwang,fandongmeng,yunlonliang,withtomzhou}@tencent.com

Abstract

Many-to-many summarization (M2MS) aims to process documents in any language and generate the corresponding summaries also in any language. Recently, large language models (LLMs) have shown strong multi-lingual abilities, giving them the potential to perform M2MS in real applications. This work presents a systematic empirical study on LLMs' M2MS ability. Specifically, we first reorganize M2MS data based on eight previous domain-specific datasets. The reorganized data contains 47.8K samples spanning five domains and six languages, which could be used to train and evaluate LLMs. Then, we benchmark 18 LLMs in a zero-shot manner and an instruction-tuning manner. Fine-tuned traditional models (*e.g.*, mBART) are also conducted for comparisons. Our experiments reveal that, zero-shot LLMs achieve competitive results with fine-tuned traditional models. After instruct-tuning, open-source LLMs can significantly improve their M2MS ability, and outperform zero-shot LLMs (including GPT-4) in terms of automatic evaluations. In addition, we demonstrate this task-specific improvement does not sacrifice the LLMs' general task-solving abilities. However, as revealed by our human evaluation, LLMs still face the factuality issue, and the instruction tuning might intensify the issue. Thus, how to control factual errors becomes the key when building LLM summarizers in real applications, and is worthy to be noted in future research.

1 Introduction

Many-to-Many Summarization (M2MS) is proposed to generate a brief summary in any language given a document also in any language (c.f., Figure 1) (Wang et al., 2023c; Bhattacharjee et al., 2023). This task is extremely challenging since it requires the ability to summarize and translate across many languages. Meanwhile, many LLMs adopt the multi-lingual setting to share the language modeling across various languages (Touvron et al., 2023a; OpenAI, 2023), making it possible to become an advanced M2MS solver in theory. However, there is still a lack of practice in exploring LLMs' M2MS performance.

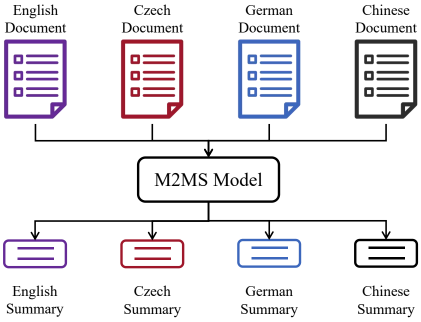

Figure 1: Illustration of many-to-many summarization.

In this paper, we try to investigate how LLMs can perform the M2MS task in real applications with multi-domain scenarios. Considering the limited diversity and the single-domain characteristic in each existing dataset, a single dataset cannot be directly used to benchmark LLMs in multi-domain scenarios. Thus, we first reorganize and select M2MS samples from eight existing multi-lingual summarization datasets (Ladhak et al., 2020; Fatima and Strube, 2021; Perez-Beltrachini and Lapata, 2021; Wang et al., 2022b; Chen et al., 2023; Bhattacharjee et al., 2023; Zheng et al., 2023). These datasets cover five domains, *i.e.*, news, how-to guides, encyclopedia, dialogue, and technology, allowing the transformation of shareable knowledge across different domains. During sample selection, we consider the intrinsic quality metrics (coverage, redundancy and coherence), and balance the number of samples among different languages and domains. For the testing samples, we also consider data contamination to ensure fair evaluations. After that, there are 47.8K samples used to train and evaluate models in our study.

[*] Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11328–11344
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Then, we benchmark 18 human-aligned LLMs, including open- and closed-source LLMs. We evaluate their zero-shot M2MS ability by prompting them with task-specific instruction and in-context examples. In this way, LLMs can leverage their instruction-following ability to perform M2MS without any parameter updating. Besides, we fine-tune two state-of-the-art traditional models (Tang et al., 2021; Wang et al., 2023c) for comparison. Our experiments reveal that the zero-shot LLMs could achieve competitive results with fine-tuned traditional models, showing the promising task-solving ability of LLMs. Furthermore, we train the open-source LLMs to perform M2MS via instruction tuning. We find that open-source LLMs can significantly improve their M2MS ability through instruct-tuning, and outperform the original models and traditional models by a large margin. Some tuned LLMs can even outperform zero-shot GPT-4 in terms of automatic metrics. In addition, we evaluate the original LLMs and the instruction-tuned LLMs on MMLU (Hendrycks et al., 2021). The results demonstrate the improvement brought by instruction-tuning does not sacrifice the LLMs' general task-solving abilities.

Moreover, as revealed by recent work, hallucination is an obstacle when building LLMs in real applications (Zhang et al., 2023). We conduct fine-grained human evaluation to figure out whether the generated summaries involve factual errors. The results indicate that open-source LLMs have more factual errors than zero-shot GPT-4. Besides, the instruction tuning on LLMs might intensify factual errors, and make LLMs tend to generate hallucinations. This issue might come from the hallucination signals of ground truth references in existing summarization datasets (Wang et al., 2022a; Gao et al., 2023). Therefore, future work should strengthen the factual consistency when building LLM summarizers in real M2MS applications.

Our main contributions are concluded as follows:

- To our knowledge, we are the first to investigate how LLMs perform the M2MS task. To this end, we reorganize and select samples from previous multi-lingual summarization datasets to construct a multi-domain M2MS scenario.
- We conduct extensive studies on 18 LLMs. The evaluation process involves zero-shot prompting and instruction tuning. Fine-tuned traditional models are also conducted for comparison.
- In-depth analyses of the M2MS results on automatic evaluation and human evaluation provide a deeper understanding of the M2MS task-solving situations in the LLM era.

2 Related Work

2.1 Summarization in Multi-Lingual World.

To adapt text summarization to the multilingual world, the summarization research field proposes the following three branch tasks:

(1) Cross-lingual summarization (CLS) aims to generate a target-language summary for a document in a *different* source language (Wang et al., 2022c). Early work typically focuses on pipeline methods (Leuski et al., 2003; Orăsan and Chiorean, 2008). Some recent studies have demonstrated that such pipeline methods suffer from error propagation and inference latency, and their performance is worse than the end-to-end ones (Zhu et al., 2019; Perez-Beltrachini and Lapata, 2021). Meanwhile, with the availability of large-scale CLS datasets, many researchers shift the research attention to end-to-end CLS, using different techniques to deal with CLS, *i.e.*, multi-task learning (Cao et al., 2020a; Bai et al., 2022; Liang et al., 2022a), knowledge distillation (Duan et al., 2019; Nguyen and Tuan, 2022) and different pre-training strategies (Xu et al., 2020; Chi et al., 2021; Wang et al., 2022b).

(2) Multi-lingual summarization (MLS) aims to process documents in multiple languages and generate their summaries in the *corresponding* language. Recently, large-scale MLS datasets (Scialom et al., 2020; Hasan et al., 2021; Liang et al., 2023) have been proposed one after another, facilitating further research on MLS. MultiSumm (Cao et al., 2020b) explores various knowledge-sharing strategies to train a MLS model among different languages. CALMS (Wang et al., 2021) proposes to train MLS models in the contrastive learning framework to share salient information extractive ability across different languages. Some studies (Aharoni et al., 2023; Qiu et al., 2023) aim to enhance the factual consistency of MLS models.

(3) Many-to-many summarization (M2MS) combines CLS and MLS into a more general setting that requires the model to summarize documents in any source language to a target language also in any language. CrossSum (Bhattacharjee et al., 2023) studies M2MS in the news domain, and shows M2MS model consistently outperforms CLS models, verifying the practicality of M2MS. Wang et al. (2023c) propose PISCES, a pre-trained M2MS model based on mBART. Besides, they explore

the summarization models trained with the settings of CLS, MLS and M2MS, and demonstrates the superiority of M2MS that allows task knowledge sharing across all languages. Different from existing work which generally uses traditional models, such as mBART (Liu et al., 2020), we first explore how well existing LLMs can perform M2MS in the zero-shot and the instruction-tuning manners.

2.2 Large Language Models.

Recently, there has been growing interest in LLMs for various NLP tasks (Zhao et al., 2023). A remarkable progress is the launch of ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023). LLMs show their powerful ability that serve as a general-purpose language task solver. Many powerful LLMs are proposed one after another to facilitate the LLM research, including LLaMa (Touvron et al., 2023a), LLaMa-2 (Touvron et al., 2023b), BaiChuan (Yang et al., 2023), Qwen (Bai et al., 2023; Yang et al., 2024), Vicuna (Chiang et al., 2023) and InternLM (Team, 2023). These LLMs generally adopt a three-stage training paradigm which first uses the next token prediction to learn the language modeling ability, and then leverages instruction tuning to enhance the model ability of following human instructions. Finally, an optional reinforcement learning with human feedback (RLHF) stage aligns LLMs' values with humans.

3 Data

Dataset Selection. To ensure the data quality, the selected multi-lingual summarization datasets should meet the following requirements: (i) the datasets should be peer-reviewed and published; (ii) the datasets should provide cross-lingual alignments for their documents and summaries across different languages to support M2MS.[1] After carefully comparing existing data, we finally choose the following eight datasets: (1) CrossSum (Bhattacharjee et al., 2023) is a news-domain dataset that collects document-summary pairs from the BBC news website. (2) XWikis (Perez-Beltrachini and Lapata, 2021) is an encyclopedia-domain dataset that collects summarization samples from Wikipedia. (3) XSAMSum (Wang et al., 2022b), (4) XMediaSum (Wang et al., 2022b) and (5) DialogSumX (Chen et al., 2023) are three dialogue-domain datasets, which are collected by manually translating the summaries of existing English dialogue summarization datasets into other languages. (6) WikiLingua (Ladhak et al., 2020) is a multi-lingual dataset in the domain of how-to guides. This dataset is collected from the WikiHow website. (7) Perseus (Zheng et al., 2023) is a technology-domain dataset that collects Chinese scientific articles with the corresponding Chinese and English summaries. (8) Spektrum (Fatima and Strube, 2021) is also a technology-domain dataset. This dataset collects samples from Spektrum der Wissenschaf (a German scientific journal).

Considering the involved languages of the chosen datasets, we make the used data support English (abbr. En), Czech (Cs), German (De), French (Fr), Chinese (Zh) and Ukrainian (Uk).

Intrinsic Metrics. After determining the datasets and the languages, we select M2MS samples from these datasets to use in our empirical study. Since the samples from a single dataset might be mixed-quality, we follow Grusky et al. (2018); Bommasani and Cardie (2020) and filter out low-quality samples based on three intrinsic quality metrics, *i.e.*, coverage, redundancy and coherence. These metrics are all automatically calculated based on the text features of document-summary pairs. For more details of these metrics and the filtering thresholds, please refer to Appendix A.

Size and Contamination Controlling. To ensure the data diversity in our empirical study, given a dataset and a specific source-target language pair, we decide to randomly select a few hundred samples from the remaining samples. Following the success of the instruction tuning in LLaMa-2 (Touvron et al., 2023b) and Vicuna (Chiang et al., 2023), we control the number of training set to tens of thousands. During sample selection, we also consider the balance of each language as well as each domain. We make the selected data contain 19,530, 14,150 and 14,150 samples in the training, validation and testing sets. As LLMs are pre-trained on massive data, their downstream performances might be inflated due to data contamination (Dong et al., 2024; Golchin and Surdeanu, 2024). To alleviate this issue, during the selection of the testing samples, we follow Golchin and Surdeanu (2024) to calculate instance-level contamination for each M2MS sample, and control the proportion of contaminated samples is less than 1% in the testing set (more details are provided in Appendix B).

Data Statistics. As shown in Table 1, the final

[1] Note that not all datasets provide these alignments, some datasets only provide monolingual document-summary pairs in multiple languages, and thus do not support summarizing documents from a language into other languages.

Tgt Src	En	Cs	De	Fr	Zh	Uk
En	3,900	1,200	3,400	1,350	3,050	1,450
Cs	1,200	1,000	1,200	1,200	1,200	-
De	1,900	1,200	2,000	1,200	1,200	-
Fr	1,400	1,200	1,200	1,050	1,165	700
Zh	2,550	1,200	1,200	1,165	2,550	1,100
Uk	1,000	-	-	700	1,000	1,000

Table 1: The number of samples w.r.t different source-target language pairs. "*Src*" and "*Tgt*" denote the source and the target languages, respectively.

LLM	Para.	Max Len.	Flores
gpt-4o-0816	-	16K	**29.1**
gpt-4-1106	-	16K	27.7
gpt-3.5-turbo-1106	-	16K	22.0
LLaMa-2-13B-chat	13B	4K	6.2
LLaMa-2-7B-chat	7B	4K	5.2
LLaMa-3-8B-chat	8B	8K	9.4
Vicuna-13B-v1.5	13B	4K	7.2
Vicuna-13B-v1.5-16k	13B	16K	6.8
Vicuna-7B-v1.5	7B	4K	6.1
Vicuna-7B-v1.5-16k	7B	16K	5.9
Baichuan2-13B-Chat	13B	4K	12.6
Baichuan2-7B-Chat	7B	4K	11.0
Qwen-14B-Chat	14B	8K	17.1
Qwen-7B-Chat	7B	8K	13.2
Qwen2.5-14B-Chat	14B	32K	19.2
Qwen2.5-7B-Chat	7B	32K	15.3
Internlm2-chat-20B	20B	32K	16.9
Internlm2-chat-7B	7B	32K	15.2

Table 2: Comparisons among LLMs used in experiments, including their parameters (Para.), maximum support length (Max Len.), and multi-lingual performance on Flores.

data covers most language pairs among the six languages except for Cs↔Uk and De↔Uk. For more details, including the number of samples w.r.t each subset, the data sources w.r.t each language pair, length distribution and domain distribution, please refer to Appendix C.

4 Experimental Setup

4.1 Evaluation LLMs

We conduct experiments on the following three types of backbones, *i.e.*, traditional models, closed- and open-source LLMs.

Traditional multi-lingual models. (1) mBART-50 (Tang et al., 2021) is a pre-trained multi-lingual model with transformer encoder-decoder architecture (Vaswani et al., 2017) and 610M parameters. (2) PISCES (610M) (Wang et al., 2023c) is an M2MS pre-trained model that extends mBART-50 by further pre-training.

Closed-source LLMs. (1) GPT-3.5-turbo (ChatGPT) (OpenAI, 2022) is created by fine-tuning a GPT-3.5 series model via reinforcement learning from human feedback (RLHF). We use *gpt-3.5-turbo-1106* in our experiments. (2) GPT-4 (OpenAI, 2023) is another advanced LLM that exhibits human-level performance on various benchmark datasets. We use *gpt-4-1106* in our experiments. (3) GPT-4o (OpenAI, 2024) is an autoregressive omni model, which accepts multi-modal inputs and can generate multi-modal outputs. The model shows superiority performance in various NLP benchmarks. We use *gpt-4o-2024-08-16* in our experiments.

Open-source LLMs. (1) LLaMa (Touvron et al., 2023b; Dubey et al., 2024) is a LLM family, which also shows remarkable performance as a general task solver. We use *LLaMa-2-7B-chat*, *LLaMa-2-13B-chat* and *LLaMa-3-8B-chat* in experiments. (2) Vicuna (Chiang et al., 2023) is created by fine-tuning LLaMa-series models on user-shared conversations collected from ShareGPT. Vicuna also provides 16k versions to support long text. We use *Vicuna-7B-v1.5*, *Vicuna-7B-v1.5-16k*, *Vicuna-13B-v1.5* and *Vicuna-13B-v1.5-16k* in experiments. (3) BaiChuan-2 (Yang et al., 2023) is also trained with instruction-tuning and RLHF. This model shows its superior multi-lingual abilities in downstream tasks. We use *Baichuan2-7B-Chat* and *Baichuan2-13B-Chat* in experiments. (4) Qwen (Bai et al., 2023; Yang et al., 2024) is a LLM family, which shows great performance as a general task solver. We use *Qwen-7B-Chat*, *Qwen-14B-Chat*, *Qwen2.5-7B-Chat* and *Qwen2.5-14B-Chat* in experiments. (5) InternLM (Team, 2023) is a multi-lingual LLM pre-trained on multi-lingual corpora. We use *Internlm2-chat-7B* and *Internlm2-chat-20B* in experiments.

To provide a deeper understanding of the above LLMs, Table 2 compares their multi-lingual performance on Flores-101 (Goyal et al., 2021), parameters and maximum support lengths. To evaluate LLMs on M2MS, there are two settings should be considered: (1) In the zero-shot prompting setting, LLMs directly perform M2MS based on a carefully designed prompt with task-specific instruction and in-context examples (Appenidx D). (2) In the instruction-tuning setting, the training samples will be used for tuning open-source LLMs. The above prompt is also used to formulate the M2MS sample into an instruction-response format.

4.2 Evaluation Metrics

We adopt ROUGE-1 (**R1**), ROUGE-2 (**R2**), ROUGE-L (**RL**) (Lin, 2004) and BERTScore (**BS**) (Zhang et al., 2020). The ROUGE scores measure the lexical overlap between the generated summaries and the references. BERTScore measures the similarity between them from a semantics perspective. Besides, following Wang et al. (2023a); Liu et al. (2024), we prompt GPT-4o to score the generated summaries in terms of conciseness (**Con.**), coherence (**Coh.**), and relevance (**Rel.**) on a 5-point scale (more details are given in Appendix E.1).

4.3 Implementation Details

For all LLMs, including open- and closed-source LLMs, we use the sampling decoding strategy, and follow Liu et al. (2023) to set the temperature to 0.1. Besides, the maximum generation length is set to 400 tokens (the length of more than 97.8% of summaries in used data is less than 400 tokens). Since different LLMs have different maximum support lengths, we truncate the input source-language document to ensure the input length is within 3600 tokens to ensure fair comparison. Please refer to Appendix E.2 for more details of utilized model checkpoints, instruction-tuning LLMs, fine-tuning traditional models, and training hours.

5 Results and Analyses

Table 3 shows the experimental results in terms of R1, RL and BS. For the full results including R2, please refer to Appendix F. We analyze the results from the following aspects:

Comparing traditional models with LLMs. When comparing traditional multi-lingual language models (mBART-50 and PISCES) with zero-shot LLMs, the fine-tuned traditional models slightly outperform the best zero-shot LLM (*i.e.*, GPT-4o). For example, GPT-4o achieves 26.0 R1 and 66.7 BS scores in overall performance, while the counterparts of mBART-50 are 27.4 and 67.8. PISCES outperforms mBART-50, and achieves 30.8 R1 and 68.6 BS scores. This finding is consistent with previous exploration of other summarization tasks (Wang et al., 2023c; Qin et al., 2023). The fine-tuned models learn the mapping from documents to summaries based on the whole training samples. In contrast, zero-shot LLMs only know a few in-context examples when performing M2MS. Without parameter updating, zero-shot LLMs could achieve competitive results with fine-tuned traditional models, showing their powerful instruction-following and in-context learning abilities.

Comparing traditional models with instruction-tuned LLMs, we find that instruction-tuned LLMs generally outperform the best traditional language model (*i.e.*, PISCES) by a large margin. For example, the instruction-tuned Vicuna-13B-16k outperforms PISCES by 7.2 R1, 7.4 RL and 5.5 BS scores. We analyze this phenomenon from the following aspects: (1) The LLMs involve more parameters than the traditional models, thus having a more powerful ability to fit the tasks-specific data with small-scale instruction samples. As shown in previous work (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021; Bhattacharjee et al., 2023), traditional models need a large number of multi-lingual summarization samples to learn how to generate a target-language summaries for the given source-language documents. Even in a single domain, traditional models generally need more than 100K samples during their training stage (Liang et al., 2022b). However, our study only uses 19.5K training samples from five domains, resulting in a great challenge to traditional models. (2) Another important aspect is the model's maximum support length. Many documents in M2MS samples contain more than 2K tokens which is larger than the maximum support length of traditional models (c.f., Appendix C). For example, mBART-50 and PISCES only support input text within 1K tokens due to the limitation in their vanilla $\mathcal{O}(n^2)$ self-attention mechanism (Vaswani et al., 2017). In contrast, LLMs typically adopt RoPE (Su et al., 2024) that comes with valuable properties such as the flexibility of being expanded to any sequence length, and the capability of equipping the linear self-attention with relative position encoding. In this manner, LLMs could support long-document inputs and are more practical in real-world scenes.

Comparisons among zero-shot LLMs. Among all zero-shot LLMs, GPT-4o achieves the best results in overall performance while GPT-4 achieves the second results in most cases. Compared with other LLMs, GPT-4o shows its powerful ability to follow human instructions to perform MSMS and generally outperforms other LLMs. Among open-source LLMs, Vicuna-13B-16k works best and reaches 22.9 R1, 13.9 RL and 66.0 BS scores in overall performance, verifying the effectiveness of instruction-tuning LLaMa-series LLMs with the ShareGPT's user conversations. Comparing Vicuna-series LLMs with other open-source LLMs, we find that though the multi-lingual ability of Vicuna-series LLMs is less than others (as demonstrated in Table 2), Vicuna-series LLMs typically

LLM	**Overall** (R1 / RL / BS)	News (R1 / RL / BS)	Encyc. (R1 / RL / BS)	Dialogue (R1 / RL / BS)	Guide (R1 / RL / BS)	Tech. (R1 / RL / BS)
Setting 1: Zero-Shot LLMs						
GPT-4o	**26.0 / 16.6 / 66.7**	**19.8 / 12.9 / 66.8**	27.9 / **16.3 / 66.0**	**29.5 / 22.1 / 70.4**	**25.1 / 16.1 / 69.0**	**34.2** / 19.1 / 69.1
GPT-4	25.7 / 16.4 / 66.4	19.5 / 12.5 / 65.9	26.9 / 14.5 / 64.8	28.9 / 21.6 / 70.0	24.0 / 15.5 / 68.4	33.8 / 18.8 / 68.9
GPT-3.5-turbo	25.2 / 16.1 / **66.7**	19.3 / 12.4 / 66.5	**28.1** / 16.1 / 65.8	24.0 / 18.5 / 66.4	22.4 / 14.6 / 67.9	33.6 / **19.3 / 69.2**
LLaMa-2-13B	21.5 / 13.0 / 64.1	17.9 / 11.8 / 65.1	25.5 / 14.6 / 63.8	19.3 / 13.8 / 64.0	18.4 / 11.3 / 64.3	30.7 / 17.2 / 64.5
LLaMa-2-7B	18.2 / 10.8 / 63.3	14.2 / 09.0 / 64.0	22.7 / 12.7 / 62.4	17.4 / 12.7 / 62.6	15.0 / 09.3 / 63.6	23.6 / 13.8 / 62.5
LLaMa-3-8B	19.5 / 12.4 / 63.5	14.9 / 09.5 / 64.6	23.3 / 13.3 / 62.8	18.0 / 13.1 / 62.8	15.7 / 10.0 / 64.2	24.1 / 14.4 / 63.0
Vicuna-13B	22.4 / 13.4 / 65.5	18.5 / 11.8 / 65.1	25.9 / 14.9 / 64.9	22.5 / 16.5 / 65.9	18.8 / 11.9 / 64.8	32.5 / 18.0 / 68.6
Vicuna-13B-16k	22.9 / 13.9 / 66.0	19.0 / 11.9 / 65.3	27.2 / 15.5 / 65.3	22.6 / 17.1 / 66.1	20.3 / 12.9 / 65.9	33.0 / 19.2 / 69.1
Vicuna-7B	22.3 / 13.7 / 65.0	17.8 / 11.6 / 65.5	26.0 / 15.4 / 64.9	22.1 / 16.2 / 67.1	18.5 / 11.3 / 65.3	31.1 / 17.5 / 67.4
Vicuna-7B-16k	22.8 / 14.1 / 65.3	18.3 / 12.0 / 65.8	27.0 / 15.1 / 65.1	21.6 / 16.2 / 66.1	19.2 / 11.7 / 64.5	31.9 / 18.2 / 66.7
Baichuan2-13B	20.5 / 12.8 / 65.0	15.9 / 10.2 / 64.9	24.4 / 13.7 / 64.1	19.8 / 15.3 / 64.9	18.1 / 11.1 / 65.0	30.0 / 16.8 / 66.2
Baichuan2-7B	20.8 / 13.2 / 65.1	16.5 / 10.5 / 65.3	24.6 / 14.1 / 64.2	21.4 / 16.2 / 66.0	17.8 / 11.1 / 64.9	30.1 / 16.1 / 64.8
Qwen-14B	21.6 / 13.0 / 65.2	17.9 / 11.5 / 65.6	25.3 / 14.3 / 64.7	21.8 / 16.3 / 66.5	18.1 / 11.1 / 64.7	32.0 / 17.8 / 64.8
Qwen-7B	21.8 / 13.1 / 64.9	18.3 / 11.5 / 66.0	25.9 / 15.1 / 65.1	21.3 / 15.9 / 66.4	17.8 / 10.9 / 65.2	30.8 / 17.8 / 65.6
Qwen2.5-14B	22.1 / 13.1 / 65.4	18.4 / 11.7 / 65.8	25.8 / 14.8 / 65.2	22.0 / 16.6 / 66.8	18.5 / 11.6 / 64.9	32.6 / 18.1 / 65.2
Qwen2.5-7B	21.9 / 13.3 / 65.1	18.6 / 11.8 / 66.5	26.5 / 15.4 / 65.4	21.9 / 16.1 / 66.6	18.1 / 10.9 / 65.6	30.9 / 18.0 / 65.7
Internlm2-20B	19.2 / 12.0 / 62.9	14.9 / 09.6 / 62.7	24.0 / 13.9 / 63.8	11.6 / 08.8 / 59.1	16.2 / 10.1 / 63.0	30.6 / 17.5 / 66.6
Internlm2-7B	18.5 / 11.6 / 62.2	14.3 / 09.5 / 62.6	23.9 / 13.3 / 63.2	11.6 / 09.1 / 58.4	16.2 / 09.9 / 62.4	29.7 / 17.7 / 64.1
Setting 2: Fine-Tuned Traditional Multi-Lingual Language Models						
mBART-50	27.4 / 19.9 / 67.8	**27.2 / 20.1** / 67.8	26.6 / 20.1 / 65.3	32.9 / 24.9 / **71.0**	25.8 / 19.5 / 68.1	23.2 / 16.7 / 65.4
PISCES	**30.8 / 22.8 / 68.6**	**27.2** / 19.8 / **68.7**	**28.2 / 20.9 / 66.0**	**34.1 / 26.8** / 70.9	**36.3 / 28.8 / 71.9**	**24.3 / 17.4 / 65.7**
Setting 3: Instruction-Tuned LLMs						
LLaMa-2-13B	37.7 / 29.4 / **74.4**	**37.1** / 27.2 / 74.2	40.2 / 32.2 / 74.2	40.3 / 32.4 / 75.4	33.0 / 26.6 / 73.4	38.2 / 26.2 / 73.4
LLaMa-2-7B	35.5 / 27.0 / 73.0	34.9 / 26.5 / 73.2	37.6 / 29.2 / 73.2	37.9 / 30.8 / 75.0	31.8 / 25.9 / 72.6	37.8 / 25.7 / 72.7
LLaMa-3-8B	36.2 / 27.5 / 73.4	35.4 / 26.9 / 73.4	37.9 / 29.8 / 73.9	38.6 / 31.5 / 75.7	32.5 / 26.4 / 73.2	38.5 / 26.2 / 73.3
Vicuna-13B	37.3 / 28.7 / 73.9	36.3 / 28.0 / 73.6	39.8 / 32.3 / **74.4**	40.4 / 32.3 / 75.9	34.2 / 28.1 / 73.5	38.4 / **26.6** / 73.7
Vicuna-13B-16k	**38.0 / 30.2** / 74.1	36.9 / **28.6 / 74.7**	**40.4 / 32.9** / 74.2	**41.2** / 33.6 / 75.9	**34.5** / 28.5 / 73.9	38.3 / 26.4 / 73.5
Vicuna-7B	35.6 / 28.0 / 73.1	34.5 / 26.2 / 73.8	38.3 / 29.1 / 73.3	38.9 / 32.3 / 75.4	31.2 / 25.7 / 72.9	36.8 / 24.5 / 72.4
Vicuna-7B-16k	36.2 / 28.6 / 73.7	35.5 / 27.7 / 73.4	38.7 / 30.3 / 74.2	38.9 / 31.8 / 75.3	32.3 / 25.9 / 72.2	37.7 / 26.5 / 73.3
Baichuan2-13B	36.1 / 28.0 / **74.4**	35.9 / 26.4 / 73.1	38.0 / 29.8 / 73.2	40.7 / 34.1 / 75.9	33.5 / 25.7 / 73.7	39.2 / 25.3 / 73.8
Baichuan2-7B	35.0 / 27.4 / 73.5	35.4 / 26.6 / 73.0	37.3 / 29.4 / 73.3	38.8 / 31.5 / 74.9	31.8 / 23.9 / 72.6	38.1 / 25.9 / 73.7
Qwen-14B	37.1 / 28.4 / 74.2	36.0 / 26.8 / 73.2	38.4 / 30.8 / 73.4	40.2 / 33.4 / 75.5	34.4 / 28.5 / 74.4	39.1 / 26.4 / 74.4
Qwen-7B	34.8 / 26.8 / 73.2	33.5 / 25.0 / 72.5	36.1 / 27.6 / 73.0	38.9 / 31.5 / 75.2	33.1 / 25.7 / 73.2	37.0 / 24.3 / 73.0
Qwen2.5-14B	37.8 / 29.3 / 74.3	36.7 / 28.0 / 74.0	39.4 / 31.6 / 73.9	40.8 / 33.6 / 75.8	34.4 / **28.8 / 74.7**	**39.4 / 26.6 / 74.8**
Qwen2.5-7B	35.2 / 27.3 / 73.7	34.2 / 25.6 / 72.9	36.6 / 28.2 / 73.7	39.5 / 32.3 / 76.0	33.4 / 26.5 / 73.6	37.7 / 24.9 / 73.6
Internlm2-20B	36.7 / 28.2 / 73.7	35.0 / 25.6 / 73.0	38.0 / 29.0 / 72.8	41.1 / **34.2 / 76.2**	**34.5** / 27.9 / 74.3	39.3 / 25.7 / 73.4
Internlm2-7B	35.7 / 27.2 / 73.5	34.1 / 25.6 / 72.6	36.2 / 28.4 / 72.8	40.7 / 33.5 / 75.9	33.3 / 27.0 / 73.3	37.8 / 25.6 / 73.0

Table 3: Experimental results of the overall performance and fine-grained results in each domain. The **bold** denotes the best performance under each setting. Encyc.: Encyclopedia; Tech.: Technology.

outperform others in M2MS, showing its superior instruction-following and in-context learning abilities distilled from ShareGPT. In addition, we also find that Vicuna-13B-16k and Vicuna-7B-16k outperform Vicuna-13B and Vicuna-7B, respectively. Though the input length is truncated to 4K tokens for all LLMs, the Vicuna-16k models scale the support length from 4K to 16K and achieve better behaviors when processing long documents.

Comparisons among instruction-tuned LLMs. Vicuna-13B-16k also performs best among all instruction-tuned LLMs, and achieves 38.0 R1, 30.2 RL and 74.1 BS scores in overall performance. Qwen2.5-14B and LLaMa-2-13B achieve promising results following closely behind Vicuna-13B-16k. Besides, all instruction-tuned LLMs significantly outperform the fine-tuned traditional models as well as zero-shot LLMs by a large margin. This finding demonstrates that LLMs can improve their task-specific ability via instruction-tuning on small-scale task data. In addition to domain-wise performance, we also show the instruction-tuned LLM performance when using a specific source or target language during evaluation. The results are provided in Figure 2. Different LLMs might be good at different languages. For example, Vicuna-13B-16k performs best when processing English and Czech documents, while Qwen-14b performs best in German and Chinese documents. As for target languages, the LLaMa and Vicuna LLMs

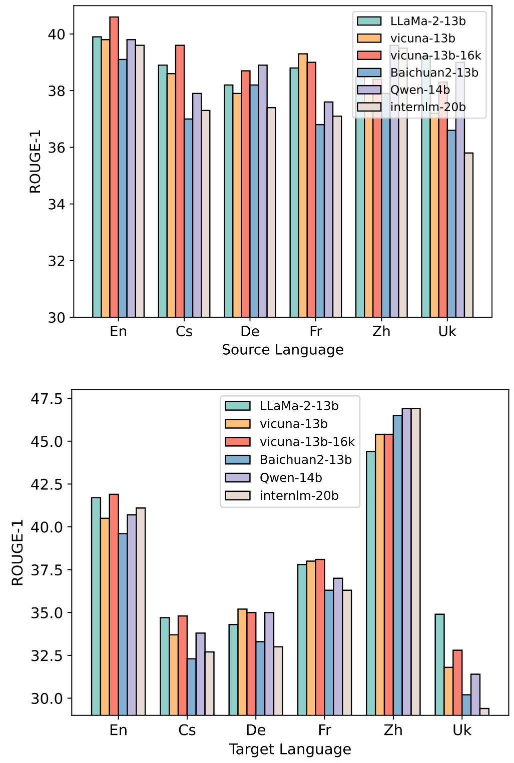

Figure 2: Language-wise performance of tuned LLMs.

are good at generating English, Czech, French and Ukrainian summaries. Baichuan2-13B, Qwen-14B and Internlm-20B do well in generating Chinese summaries.

LLM evaluation results. Table 4 shows the evaluation results using GPT-4o evaluation. The instruction-tuned LLMs significantly outperform traditional models in all metrics. For zero-shot LLMs, they tend to generate relatively lengthy summaries compared to traditional models or fine-tuned LLMs, resulting in low conciseness. This is because the alignment phase in LLMs emphasizes the usefulness of the models, making them provide detailed information. After tuning, LLMs can study to generate short summaries, and significantly improve their conciseness scores. In terms of coherence and relevance, instruction-tuning also brings improvements to LLMs. Moreover, we find that some tuned LLMs (*e.g.*, LLaMa-2-7B and Vicuna-7B) achieve lower coherence and relevance than zero-shot GPT-4, though they outperform GPT-4 in terms of ROUGE scores and BERTScore.

The effects of training scales. In our main experiments, we use 19.5K training samples to fine-tune LLMs and traditional models. We further discuss the effects of the training scales on model performance, please refer to Appendix G.

	Zero-shot Results				Ins-tuned Results		
	Con.	Coh.	Rel.		Con.	Coh.	Rel.
GPT-4o	3.30	**4.66**	**4.83**				
GPT-4	3.27	4.63	4.75	mBART-50	3.82	3.42	4.02
GPT-3.5-turbo	3.04	4.56	4.69	PISCES	3.95	3.58	4.15
LLaMa-2-13B	3.20	4.34	4.50	LLaMa-2-13B	4.59	4.70	**4.72**
LLaMa-2-7B	3.28	4.29	4.39	LLaMa-2-7B	4.42	4.57	4.65
LLaMa-3-8B	3.32	4.31	4.48	LLaMa-3-8B	4.50	4.59	4.68
Vicuna-13B	3.15	4.35	4.51	Vicuna-13B	**4.70**	4.68	4.70
Vicuna-13B-16k	3.10	4.39	4.52	Vicuna-13B-16k	4.63	**4.71**	4.66
Vicuna-7B	3.06	4.27	4.45	Vicuna-7B	4.47	4.55	4.62
Vicuna-7B-16k	3.10	4.30	4.37	Vicuna-7B-16k	4.53	4.58	4.60
Baichuan2-13B	**3.37**	4.34	4.46	Baichuan2-13B	4.51	4.65	4.69
Baichuan2-7B	3.19	4.25	4.29	Baichuan2-7B	4.40	4.51	4.53
Qwen-14B	3.25	4.32	4.40	Qwen-14B	4.55	4.64	4.60
Qwen-7B	3.30	4.23	4.33	Qwen-7B	4.39	4.51	4.45
Qwen2.5-14B	3.32	4.37	4.48	Qwen2.5-14B	4.63	4.69	4.65
Qwen2.5-7B	3.27	4.29	4.32	Qwen2.5-7B	4.42	4.56	4.50
Internlm2-20B	3.17	4.29	4.37	Internlm2-20B	4.51	4.60	4.67
Internlm2-7B	3.04	4.18	4.28	Internlm2-7B	4.25	4.47	4.52

Table 4: Overall performance in terms of GPT-4o evaluation (Ins-tuned: Instruction-tuned).

Model	Rate	Model	Rate
LLaMa-2-13B	91.1† / 98.7‡	Baichuan2-13B	98.7† / 99.4‡
LLaMa-2-7B	81.6† / 98.7‡	Baichuan2-7B	94.8† / 99.4‡
LLaMa-3-8B	96.6† / 98.8‡	Qwen-14B	98.7† / 99.3‡
Vicuna-13B	96.6† / 98.6‡	Qwen-7B	98.7† / 99.1‡
Vicuna-13B-16k	98.7† / 98.8‡	Qwen2.5-14B	99.2† / 99.4‡
Vicuna-7B	94.5† / 99.2‡	Qwen2.5-7B	98.5† / 99.1‡
Vicuna-7B-16k	97.5† / 98.7‡	Internlm2-20B	97.0† / 98.7‡
		Internlm2-7B	97.2† / 98.5‡

Table 5: Correct language rate (%) of the summaries († and ‡ denote the results of zero-shot and instruction-tuned LLMs, respectively).

6 Discussion

Are generated summaries in the correct language? As reported by Wang et al. (2023c), using traditional models as multi-lingual summarizers might generate in the wrong languages instead of the given target language. We wonder whether LLM-generated summaries are in the correct language. To this end, we use *fastlangid*[2] to detect the generated summaries, and calculate the rate of the generated summaries in the correct language, named correct language rate (**CR**). As shown in Table 5, we find that most zero-shot LLMs could follow the language requirements in the prompt and generate summaries in the right target language with ≥ 95 CR score. This is because the LLMs are trained with multi-lingual corpora that include a large number of parallel sentences across different languages, making the model already learn the multi-lingual skills. After tuning, LLMs generally improve their CR scores. Thus, using LLMs to serve as the backbones of the M2MS systems has great potential in real-world applications.

[2] https://pypi.org/project/fastlangid/

LLM	Before	After	LLM	Before	After
LLaMa-2-13B	53.5	54.3	Baichuan2-13B	54.5	54.1
LLaMa-2-7B	47.2	47.3	Baichuan2-7B	52.9	52.0
Vicuna-13B	55.5	55.8	Qwen-14B	66.0	65.9
Vicuna-13B-16k	54.3	54.5	Qwen-7B	56.2	57.0
Vicuna-7B	49.8	49.6	Internlm2-20B	65.0	64.6
Vicuna-7B-16k	48.0	48.4	Internlm2-7B	59.1	58.9

Table 6: The LLMs' performance on MMLU. "*Before*" and "*After*" denote the results of LLMs before and after the M2MS instruction tuning, respectively.

Does the instruction tuning on M2MS influence LLMs' general ability? As shown in Section 5, LLMs can improve their M2MS ability by instruction-tuning on the collected training samples. An important question arises naturally, *i.e.*, *does this task-specific improvement sacrifice the general task-solving ability of LLMs?* To figure out this question, we further evaluate the LLMs before and after M2MS instruction tuning on the MMLU evaluation dataset (Hendrycks et al., 2021). The MMLU dataset covers 57 tasks including elementary mathematics, US history, computer science, law, etc, and is designed to evaluate models' world knowledge and problem-solving ability. We follow previous LLM work (Touvron et al., 2023b; Yang et al., 2023; Bai et al., 2023), and adopt the 5-shot evaluation strategy. The experimental results are provided in Table 6. As we can see, the M2MS instruction tuning on LLMs does not sacrifice their general task-solving ability. Some instruction-tuned LLMs (*i.e.*, LLaMa-2-13B and Qwen-7B) even outperform their original models.

Can large model generate factually consistent summaries? As revealed by Maynez et al. (2020); Gao et al. (2023), the model-generated summaries might be inconsistent with the source documents. We want to know if LLMs can generate factually consistent summaries. To this end, we randomly select 100 samples from the testing set and conduct fine-grained human evaluation on the summaries generated by GPT-4, zero-shot & tuned LLaMa-2-13B, and zero-shot & tuned Vicuna-13B-16k. Following Gao et al. (2023), for each sample, we employ human evaluators to annotate the following four types of factual errors (if has): (1) hallucination error: a generated summary contains events not directly inferable from the given document. (2) particulars error: the summary contains the right events but some details are inaccurate or mistaken. (3) predicate error: the predicate in the summary is contradictory to the source document. (4) entity error: the entity of an event in the summary is wrong. More details about human evaluation are given in Appendix H. Table 7 reports the proportion of each error. The summaries generated by GPT-4 have the lowest error proportion in terms of all errors. Besides, among all types of factual errors, the hallucination error occurs more frequently, indicating it is a non-trivial issue when adapting LLMs as the summarizers. Another finding is that instruction tuning on LLMs might intensify their factual issue especially the hallucination and the particulars errors. We conjecture this because the summaries in the training samples are written by humans, thus they might involve more information (like background information) beyond the given documents. Such an information gap might encourage LLMs to generate hallucinations during tuning. As reported by previous studies (Wang et al., 2022a; Gao et al., 2023), the ground truth references in summarization data also have the hallucination error. Therefore, when building LLM M2MS summarizers in the real applications, this error should be carefully considered during instruction tuning.

	Hallu.	Parti.	Predi.	Entity
GPT-4	8	3	5	3
Zero-shot LLaMa-2-13B	17	13	9	12
Tuned LLaMa-2-13B	23	18	7	9
Zero-shot Vicuna-13B-16k	12	8	8	10
Tuned Vicuna-13B-16k	17	16	10	6

Table 7: Fine-grained human evaluation results on factuality (Hallu.: hallucination; Parti.: particulars; Predi.: predicate).

7 Conclusion

In this paper, we explore how well off-the-shelf LLMs can deal with the many-to-many summarization (M2MS) task. Considering the limited diversity and the single domain characteristics in each single dataset, we reorganize M2MS data based on eight existing multi-lingual summarization datasets, and the used data covers five domains and six languages. Based on it, we conduct extensive experiments on various open- and closed-source LLMs. Our results indicate that the zero-shot LLMs could achieve competitive results with fine-tuned traditional models. Furthermore, through instruction tuning, open-source LLMs can significantly improve their M2MS ability, and not sacrifice their general capabilities. However, as shown in our human evaluation, LLMs still face the factuality issue, and the instruction tuning might intensify this issue, which is worth noting in future research.

Limitations

While we evaluate the performance of LLMs on the many-to-many summarization task, there are some limitations worth noting: (1) We only evaluate the lower threshold of these models' M2MS performance. Prompts are important to guide LLMs to perform specific tasks, and future work could explore better prompts to obtain better results. (2) The used data in our empirical study only involves data from existing multi-lingual summarization datasets. Future work could extend it with more domains in the real scenes.

Ethical Considerations

In this paper, we use multiple LLMs (*e.g.*, GPT-4o, GPT-4, GPT-3.5-turbo and LLaMa-2) and traditional language models (*e.g.*, mBART) as the M2MS models in experiments. During instruction tuning and fine-tuning, the adopted M2MS samples mainly come from previous datasets, *i.e.*, CrossSum (Bhattacharjee et al., 2023), XWikis (Perez-Beltrachini and Lapata, 2021), XSAMSum (Wang et al., 2022b), XMediaSum (Wang et al., 2022b), DialogSumX (Chen et al., 2023), WikiLingua (Ladhak et al., 2020), Perseus (Zheng et al., 2023) and Spektrum (Fatima and Strube, 2021). Therefore, the trained models might involve the same biases and toxic behaviors exhibited by these datasets.

References

Roee Aharoni, Shashi Narayan, Joshua Maynez, Jonathan Herzig, Elizabeth Clark, and Mirella Lapata. 2023. Multilingual summarization with factual consistency evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3562–3591, Toronto, Canada. Association for Computational Linguistics.

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*.

Yu Bai, Heyan Huang, Kai Fan, Yang Gao, Yiming Zhu, Jiaao Zhan, Zewen Chi, and Boxing Chen. 2022. Unifying cross-lingual summarization and machine translation with compression rate. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1087–1097.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2023. CrossSum: Beyond English-centric cross-lingual summarization for 1,500+ language pairs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2541–2564, Toronto, Canada. Association for Computational Linguistics.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Yue Cao, Hui Liu, and Xiaojun Wan. 2020a. Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231, Online. Association for Computational Linguistics.

Yue Cao, Xiaojun Wan, Jinge Yao, and Dian Yu. 2020b. Multisumm: Towards a unified model for multi-lingual abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):11–18.

Yulong Chen, Huajian Zhang, Yijie Zhou, Xuefeng Bai, Yueguan Wang, Ming Zhong, Jianhao Yan, Yafu Li, Judy Li, Xianchao Zhu, and Yue Zhang. 2023. Revisiting cross-lingual summarization: A corpus-based study and a new benchmark with improved annotation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9332–9351, Toronto, Canada. Association for Computational Linguistics.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei. 2021. mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1671–1683, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12039–12050, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. 2019. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy. Association for Computational Linguistics.

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*.

Mehwish Fatima and Michael Strube. 2021. A novel Wikipedia based dataset for monolingual and cross-lingual summarization. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 39–50, Online and in Dominican Republic. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Mingqi Gao, Wenqing Wang, Xiaojun Wan, and Yuemei Xu. 2023. Evaluating factuality in cross-lingual summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12415–12431, Toronto, Canada. Association for Computational Linguistics.

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

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

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Anton Leuski, Chin-Yew Lin, Liang Zhou, Ulrich Germann, Franz Josef Och, and Eduard H. Hovy. 2003. Cross-lingual c*st*rd: English access to hindi information. *ACM Trans. Asian Lang. Inf. Process.*, 2:245–269.

Yunlong Liang, Fandong Meng, Jinan Xu, Jiaan Wang, Yufeng Chen, and Jie Zhou. 2023. Summary-oriented vision modeling for multimodal abstractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2934–2951, Toronto, Canada. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Chulun Zhou, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2022a. A variational hierarchical model for neural cross-lingual summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2088–2099, Dublin, Ireland. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Chulun Zhou, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2022b. A variational hierarchical model for neural cross-lingual summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2088–2099, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ran Liu, Ming Liu, Min Yu, He Zhang, Jianguo Jiang, Gang Li, and Weiqing Huang. 2024. SumSurvey:

An abstractive dataset of scientific survey papers for long document summarization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9632–9651, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Alignbench: Benchmarking chinese alignment of large language models. *Preprint*, arXiv:2311.18743.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thong Nguyen and Luu Anh Tuan. 2022. Improving neural cross-lingual summarization via employing optimal transport distance for knowledge distillation. *Proc. of AAAI*.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

OpenAI. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Constantin Orăsan and Oana Andreea Chiorean. 2008. Evaluation of a cross-lingual Romanian-English multi-document summariser. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Yifu Qiu, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2023. Detecting and mitigating hallucinations in multilingual summarisation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8932, Singapore. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Yuting Tang, Ratish Puduppully, Zhengyuan Liu, and Nancy Chen. 2023. In-context learning of large language models for controlled dialogue summarization: A holistic benchmark and empirical analysis. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 56–67, Singapore. Association for Computational Linguistics.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Bin Wang, Chen Zhang, Yan Zhang, Yiming Chen, and Haizhou Li. 2022a. Analyzing and evaluating faithfulness in dialogue summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4897–4908, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Danqing Wang, Jiaze Chen, Hao Zhou, Xipeng Qiu, and Lei Li. 2021. Contrastive aligned joint learning for multilingual summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2739–2750, Online. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023b. Zero-shot cross-lingual summarization via large language models. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 12–23, Singapore. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022b. ClidSum: A benchmark dataset for cross-lingual dialogue summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7729, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022c. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023c. Towards unifying multi-lingual and cross-lingual summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15127–15143, Toronto, Canada. Association for Computational Linguistics.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, et al. 2024. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Ruochen Xu, Chenguang Zhu, Yu Shi, Michael Zeng, and Xuedong Huang. 2020. Mixed-lingual pre-training for cross-lingual summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 536–541, Suzhou, China. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Shaohui Zheng, Zhixu Li, Jiaan Wang, Jianfeng Qu, An Liu, Lei Zhao, and Zhigang Chen. 2023. Long-document cross-lingual summarization. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1084–1092.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3054–3064, Hong Kong, China. Association for Computational Linguistics.

A Details of Intrinsic Metrics

Following Grusky et al. (2018); Bommasani and Cardie (2020), we filter out low-quality summarization samples based on the following three intrinsic quality metrics:

(1) *Coverage* evaluates the percentage of words in a summary that are part of an extractive fragment from the document (Grusky et al., 2018):

$$\text{Coverage}(D, S) = \frac{1}{|S|} \sum_{u \in F(D,S)} |u| \quad (1)$$

where D and S denote a document and the corresponding summary, respectively. $F(D, S)$ is the set of all extractive fragments that appear in both D and S. A smaller number of coverage indicates the summary is more abstractive.

(2) *Redundancy* measures whether sentences in a summary are similar to each other (Bommasani and Cardie, 2020). Assuming a summary S has m sentences $S = \{s_1, s_2, ..., s_m\}$, redundancy is formally calculated as follows:

$$\text{Redundancy}(S) = \frac{1}{\binom{m}{2}} \sum_{1 \leq i \leq m-1} \sum_{i+1 \leq j \leq m} \text{RL}(m_i, m_j) \quad (2)$$

where RL$(\cdot)$ denotes the ROUGE-L score (Lin, 2004). Generally, the lower the redundancy, the higher the sample quality.

(3) *Coherence* measures the semantic coherence of a summary $S = \{s_1, ..., s_m\}$ by predicting the probability of each successive sentence conditioned on the previous one using a language model (Bommasani and Cardie, 2020):

$$\text{Coherence}(S) = \frac{1}{m-1} \sum_{i=2}^{m} P_\theta(s_i | s_{i-1}) \quad (3)$$

where P_θ denotes the predicted probability of a language model. Here, we adopt mBERT (Devlin et al., 2019) as P_θ. Generally, a high coherence score indicates the high quality of the sample.

For each intrinsic metric, we set a threshold to filter low-quality samples. Specifically, inspired by Bommasani and Cardie (2020), for each sample, its coverage should be less than $\alpha_{\text{cov}} = 0.9$, the redundancy should be less than $\alpha_{\text{red}} = 0.2$, and the coherence needs to be more than $\alpha_{\text{coh}} = 0.9$. Otherwise, the sample will be filtered out.

B Data Contamination

Data contamination is a potential major issue in measuring LLMs' performance on downstream tasks, where the testing data might be in the pertaining corpora of LLMs (Xu et al., 2024). Golchin and Surdeanu (2024) propose an effective method using BLEURT & ROUGE-L metrics to measure instance-level contamination, *i.e.*, identifying if an instance (usually a sentence or document) is contaminated for a given LLM. The method can be used in both open-source and closed-source LLMs.

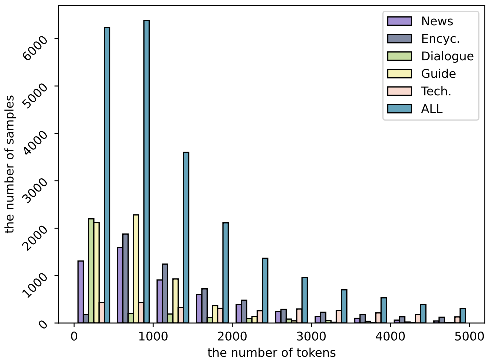

Figure 3: Length distributions of M2MS samples w.r.t different domains.

When selecting the testing samples in our empirical study, after filtering samples via intrinsic metrics (Appendix A), we calculate the instance-level contamination for each sample w.r.t GPT-4o, Vicuna-7B, Baichuan2-7B, Qwen2.5-7B-Chat, LLaMa-3-8B-chat and Internlm2-7B.[3] The uncontaminated samples will be randomly selected to form the testing set. Some directions in the domain-specific datasets might contain only a few hundred samples (*e.g.*, CrossSum only contains about 300 Fr⇒Zh samples), thus we cannot ensure data contamination and testing scale simultaneously in these directions. Therefore, an extremely small number of samples labeled as "contaminated" will also be included in our testing set, and they account for less than 1% of the whole testing set.

C Data Statistics

Language and Source Distribution. For a specific language pair, the number of samples in each subset (training, validation, and testing sets) and the corresponding data sources are provided in Table 8.

Length and Domain Distributions. To calculate the length of documents and summaries across different languages, we use tiktoken[4] to tokenize the documents and summaries, and calculate their token-level length. As shown in Table 9, the average length of source documents typically reaches thousands of tokens, while the counterpart of target summaries is within 200 tokens. From the perspective of domains, Table 10 shows the average length

[3] For the same series of LLMs, we select one to measure the contamination.

[4] https://github.com/openai/tiktoken

Src \ Tgt	En	Cs	De	Fr	Zh	Uk
En	1400 / 1250 / 1250 (CR, XW, XS, XM, DI, WI, SP)	500 / 350 / 350 (XW, WI)	1300 / 1050 / 1050 (XW, XS, XM, WI, SP)	500 / 425 / 425 (CR, XW, DI, WI)	1150 / 950 / 950 (CR, XW, XS, XM, DI, WI)	700 / 375 / 375 (CR, DI)
Cs	500 / 350 / 350 (XW, WI)	300 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	-
De	800 / 550 / 550 (XW, WI, SP)	500 / 350 / 350 (XW, WI)	900 / 550 / 550 (XW, WI, SP)	500 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	-
Fr	650 / 375 / 375 (CR, XW, WI)	500 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	300 / 375 / 375 (CR, XW, WI)	565 / 300 / 300 (CR, XW, WI)	500 / 100 / 100 (CR)
Zh	900 / 825 / 825 (CR, XW, WI, PE)	500 / 350 / 350 (XW, WI)	500 / 350 / 350 (XW, WI)	565 / 300 / 300 (CR, XW, WI)	900 / 825 / 825 (CR, XW, WI, PE)	500 / 300 / 300 (CR)
Uk	400 / 300 / 300 (CR)	-	-	400 / 150 / 150 (CR)	400 / 300 / 300 (CR)	400 / 300 / 300 (CR)

Table 8: The number of training/validation/testing samples and the data sources w.r.t different source-target language pairs. "*Src*" and "*Tgt*" denote the source and the target languages, respectively. (CR: CrossSum; XW: XWikis; XS: XSAMSum; XM: XMediaSum; DI: DialogSumX; WI: WikiLingua; PE: Perseus; SP: Spektrum)

		En→X	CS→X	De→X	Fr→X	Zh→X	Uk→X
Training	Doc.	869.05	2418.82	2111.35	1385.84	2534.56	1693.38
	Sum.	65.64	124.67	196.98	84.11	150.68	57.97
Validation	Doc.	897.45	2364.50	2087.41	1345.95	2164.52	1477.90
	Sum.	62.34	130.32	177.85	88.44	141.26	57.21
Testing	Doc.	791.15	2361.89	2074.92	1280.33	2269.90	1346.57
	Sum.	52.32	125.68	198.89	86.41	134.35	52.02
		X→En	X→Cs	X→De	X→Fr	X→Zh	X→Uk
Training	Doc.	2105.74	2204.62	2080.01	1627.83	2102.88	1050.10
	Sum.	115.40	159.83	141.19	58.17	140.67	89.05
Validation	Doc.	1731.99	2121.00	2307.03	1209.37	1821.54	982.90
	Sum.	98.52	167.79	124.08	57.89	129.91	89.89
Testing	Doc.	1769.09	1807.71	2092.57	1502.02	1883.59	799.59
	Sum.	96.41	163.11	142.47	59.25	119.95	79.27

Table 9: The token-level average length of documents (Doc.) and summaries (Sum.) in the data w.r.t different source and target languages. En→X/X→En indicates all samples whose documents/summaries are in English.

		News	Encyc.	Dialogue	Guide	Tech.
Training	Num.	4680	4650	2550	4650	3000
	Doc.	1350.21	2020.66	733.24	772.6	2857.5
	Sum.	63.36	121.46	42.04	69.96	235.14
Validation	Num.	2300	3425	2600	3425	2400
	Doc.	1101.92	1936.80	728.72	767.20	2687.42
	Sum.	57.18	122.57	40.32	67.10	221.87
Testing	Num.	2300	3425	2600	3425	2400
	Doc.	1055.44	1994.10	749.52	732.96	2882.51
	Sum.	54.72	123.86	36.18	68.36	245.10

Table 10: The token-level average length of documents (Doc.) and summaries (Sum.) in the data w.r.t different domains. "*Num.*" indicates the number of samples in each domain. Encyc.: Encyclopedia; Tech.: Technology

of documents as well as summaries w.r.t different domains. As we can see, the average document length in the encyclopedia and technology domains is generally more than that in other domains. The average length of dialogue and guide documents is less than 800 tokens, making them the shortest document length among all domains. To provide a deeper understanding of the used data in our empirical study, Figure 3 shows the length distributions of different domains.

D M2MS Prompt

Inspired by previous LLM summarization studies (Wang et al., 2023b; Tang et al., 2023) and the in-context learning technique (Dong et al., 2022; Min et al., 2022), we attempt various M2MS prompts on GPT-3.5-turbo and GPT-4, and choose the prompt with the best results (using both automatic and human evaluation) on a pilot experiment. Specifically, as shown in Figure 4, the final chosen prompt is designed with task descriptions, domain information and a few output examples. `[source language]` and `[target language]` are selected from "English", "Czech", "German", "French", "Chinese" and "Ukrainian". When the source and the target languages are the same, the content in parentheses will be omitted. `[domain]` indicates the domain of the input document, which is selected from "news", "encyclopedia", "dialogue", "how-to guides" and "technology". `[example summary i]` ($i \in \{1, 2, 3\}$) denotes a ground truth summary randomly selected from the training samples. `[document]` represents the current input document that needs to generate the corresponding summary.

System: You are a professional literary worker and proficient in [source language] (and [target language]). You are good at summarizing [source language] [domain] documents into [target language] summaries.

The following example summaries are written by you previously, and have received compliments:
Summary 1: [example summary 1]
Summary 2: [example summary 2]
Summary 3: [example summary 3]

Your task is to continue to summarize the [source language] [domain] documents into [target language] summaries. The newly generated summaries should have a similar style to your previously written summaries.

User: The [source language] document is provided as follows:

[document]

Please generate its [target language] summary.

Figure 4: Illustration of the used M2MS prompt that includes a system round and a user round.

E Implementation Details

E.1 Implementation Details of Evaluation Metrics

To calculate the ROUGE scores in the multi-lingual setting, we use the *multi-lingual rouge*[5] toolkit. For BERTScore, we use the *bert-score*[6] toolkit, and set the backbone to *bert-base-multilingual-cased*.

During evaluation via GPT-4o, the used prompt is "*I will provide you with a summary of a document. Please rate the summary on a scale of one to five in terms of conciseness, coherence, and relevance.*". We use *gpt-4o-2024-0816* as the evaluator. Since the testing set in our study contains more than 14K samples, it is a high cost to evaluate all model-generated summaries via GPT-4o. Thus, we randomly select 500 samples to conduct the LLM evaluation.

E.2 Implementation Details of LLMs

Instruction-Tuning Details. All LLMs are tuned on 8×NVIDIA A800 GPUs (80G) with 1e-5 learning rate and 32 (8×4 gradient accumulation) batch size. We follow the success of instruction tuning in LLaMa-2 (Touvron et al., 2023b), and set the training epochs to 2. We use the DeepSpeed optimization library[7], and set ZeRO-2 optimization. For Internlm2-chat-20B, we also offload the optimizer into the CPU to avoid CUDA out-of-memory error. Flash attention (v2) (Dao et al., 2022) is also employed to save memory. During tuning, documents are also truncated to ensure the input length is within 3,600 tokens to ensure fairness. For the instruction-tuned LLMs, we use the same decoding strategy as the zero-shot ones.

Model Checkpoints. (1) For traditional multilingual language models, we use mBART-50 (610M)[8] (Tang et al., 2021) and PISCES (610M)[9] (Wang et al., 2023c). (2) For open-source LLMs, we use *LLaMa-2-7B-chat*[10], *LLaMa-2-13B-chat*[11], *LLaMa-3-8B-chat*[12], *Vicuna-7B-v1.5*[13], *Vicuna-7B-v1.5-16k*[14], *Vicuna-13B-v1.5*[15], *Vicuna-13B-v1.5-16k*[16], *Baichuan2-7B-Chat*[17], *Baichuan2-13B-Chat*[18] *Qwen-7B-Chat*[19], *Qwen-14B-Chat*[20], *Qwen2.5-7B-Chat*[21], *Qwen2.5-14B-Chat*[22], *Internlm2-chat-7B*[23] and *Internlm2-chat-20B*[24] in experiments. All model checkpoints are available at the Huggingface community.

Fine-Tuning Details. To fine-tune traditional multi-lingual language models, *i.e.*, mBART-50 and PISCES, we follow Wang et al. (2023c) and set the learning rate to 3e-5, batch size to 8×8, and epochs to 10. Experiments are conducted on 8×NVIDIA A800 GPUs (80G). Different from LLMs, the source-language documents are directly input into these models without any prompts. Following previous work (Wang et al., 2023c; Bhattacharjee et al., 2023), a language tag is appended

[5] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring

[6] https://github.com/Tiiiger/bert_score

[7] https://github.com/microsoft/DeepSpeed

[8] https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

[9] https://huggingface.co/Krystalan/PISCES

[10] https://huggingface.co/meta-llama/LLaMa-2-7B-chat-hf

[11] https://huggingface.co/meta-llama/LLaMa-2-13B-chat-hf

[12] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[13] https://huggingface.co/lmsys/Vicuna-7B-v1.5

[14] https://huggingface.co/lmsys/Vicuna-7B-v1.5-16k

[15] https://huggingface.co/lmsys/Vicuna-13B-v1.5

[16] https://huggingface.co/lmsys/Vicuna-13B-v1.5-16k

[17] https://huggingface.co/baichuan-inc/Baichuan2-7B-Chat

[18] https://huggingface.co/baichuan-inc/Baichuan2-13B-Chat

[19] https://huggingface.co/Qwen/Qwen-7B-Chat

[20] https://huggingface.co/Qwen/Qwen-14B-Chat

[21] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[22] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct

[23] https://huggingface.co/internlm/Internlm2-chat-7B

[24] https://huggingface.co/internlm/Internlm2-chat-20B

LLM	**Overall** (R1 / R2 / RL / BS)	News (R1 / R2 / RL / BS)	Encyc. (R1 / R2 / RL / BS)	Dialogue (R1 / R2 / RL / BS)	Guide (R1 / R2 / RL / BS)	Tech. (R1 / R2 / RL / BS)
Setting 1: Zero-Shot LLMs						
GPT-4o	**26.0 / 12.3 / 16.6 / 66.7**	**19.8 / 09.1 / 12.9 / 66.8**	27.9 / **12.5 / 16.3 / 66.0**	**29.5 / 17.3 / 22.1 / 70.4**	**25.1 / 11.3 / 16.1 / 69.0**	**34.2 / 16.3** / 19.1 / 69.1
GPT-4	25.7 / 12.1 / 16.4 / 66.4	19.5 / 08.8 / 12.5 / 65.9	26.9 / 11.2 / 14.5 / 64.8	28.9 / 17.0 / 21.6 / 70.0	24.0 / 10.7 / 15.5 / 68.4	33.8 / 16.1 / 18.8 / 68.9
GPT-3.5-turbo	25.2 / 11.3 / 16.1 / **66.7**	19.3 / 08.8 / 12.4 / 66.5	**28.1** / 12.2 / 16.1 / 65.8	24.0 / 13.5 / 18.5 / 66.4	22.4 / 09.4 / 14.6 / 67.9	33.6 / 14.9 / **19.3 / 69.2**
LLaMa-2-13B	21.5 / 09.2 / 13.0 / 64.1	17.9 / 08.0 / 11.8 / 65.1	25.5 / 10.8 / 14.6 / 63.8	19.3 / 09.2 / 13.8 / 64.0	18.4 / 06.3 / 11.3 / 64.3	30.7 / 12.6 / 17.2 / 64.5
LLaMa-2-7B	18.2 / 06.9 / 10.8 / 63.3	14.2 / 05.6 / 09.0 / 64.0	22.7 / 08.3 / 12.7 / 62.4	17.4 / 07.4 / 12.7 / 62.6	15.0 / 04.9 / 09.3 / 63.6	23.6 / 08.2 / 13.8 / 62.5
LLaMa-3-8B	19.5 / 07.9 / 12.4 / 63.5	14.9 / 05.8 / 09.5 / 64.6	23.3 / 08.6 / 13.3 / 62.8	18.0 / 07.7 / 13.1 / 62.8	15.7 / 05.2 / 10.0 / 64.2	24.1 / 08.5 / 14.4 / 63.0
Vicuna-13B	22.4 / 08.9 / 13.4 / 65.5	18.5 / 08.4 / 11.8 / 65.1	25.9 / 10.6 / 14.9 / 64.9	22.5 / 11.1 / 16.5 / 65.9	18.8 / 07.0 / 11.9 / 64.8	32.5 / 13.7 / 18.0 / 68.6
Vicuna-13B-16k	22.9 / 09.7 / 13.9 / 66.0	19.0 / 08.2 / 11.9 / 65.3	27.2 / 11.0 / 15.5 / 65.3	22.6 / 11.5 / 17.1 / 66.1	20.3 / 08.0 / 12.9 / 65.9	33.0 / 14.7 / 19.2 / 69.1
Vicuna-7B	22.3 / 09.1 / 13.7 / 65.0	17.8 / 07.6 / 11.6 / 65.5	26.0 / 10.4 / 15.4 / 64.9	22.1 / 10.7 / 16.2 / 67.1	18.5 / 06.9 / 11.3 / 65.3	31.1 / 12.8 / 17.5 / 67.4
Vicuna-7B-16k	22.8 / 09.4 / 14.1 / 65.3	18.3 / 08.1 / 12.0 / 65.8	27.0 / 11.2 / 15.1 / 65.1	21.6 / 10.0 / 16.2 / 66.1	19.2 / 06.7 / 11.7 / 64.5	31.9 / 13.1 / 18.2 / 66.7
Baichuan2-13B	20.5 / 08.6 / 12.8 / 65.0	15.9 / 06.8 / 10.2 / 64.9	24.4 / 09.6 / 13.7 / 64.1	19.8 / 09.9 / 15.3 / 64.9	18.1 / 06.2 / 11.1 / 65.0	30.0 / 12.3 / 16.8 / 66.2
Baichuan2-7B	20.8 / 08.4 / 13.2 / 65.1	16.5 / 06.8 / 10.5 / 65.3	24.6 / 09.5 / 14.1 / 64.2	21.4 / 10.2 / 16.2 / 66.0	17.8 / 06.3 / 11.1 / 64.9	30.1 / 12.5 / 16.1 / 64.8
Qwen-14B	21.6 / 09.6 / 13.0 / 65.2	17.9 / 08.2 / 11.5 / 65.6	25.3 / 10.8 / 14.3 / 64.7	21.8 / 10.9 / 16.3 / 66.5	18.1 / 07.2 / 11.1 / 64.7	32.0 / 13.5 / 17.8 / 64.8
Qwen-7B	21.8 / 08.5 / 13.1 / 64.9	18.3 / 08.0 / 11.5 / 66.0	25.9 / 10.7 / 15.1 / 65.1	21.3 / 10.6 / 15.9 / 66.4	17.8 / 06.6 / 10.9 / 65.2	30.8 / 12.9 / 17.8 / 65.6
Qwen2.5-14B	22.1 / 09.8 / 13.1 / 65.4	18.4 / 08.4 / 11.7 / 65.8	25.8 / 11.2 / 14.8 / 65.2	22.0 / 11.1 / 16.6 / 66.8	18.5 / 07.3 / 11.6 / 64.9	32.6 / 13.8 / 18.1 / 65.2
Qwen2.5-7B	21.9 / 08.4 / 13.3 / 65.1	18.6 / 08.1 / 11.8 / 66.5	26.5 / 10.9 / 15.4 / 65.4	21.9 / 11.0 / 16.1 / 66.6	18.1 / 06.7 / 10.9 / 65.6	30.9 / 12.9 / 18.0 / 65.7
Internlm2-20B	19.2 / 07.8 / 12.0 / 62.9	14.9 / 06.5 / 09.6 / 62.7	24.0 / 10.0 / 13.9 / 63.8	11.6 / 05.4 / 08.8 / 59.1	16.2 / 06.1 / 10.1 / 63.0	30.6 / 13.9 / 17.5 / 66.6
Internlm2-7B	18.5 / 07.2 / 11.6 / 62.2	14.3 / 06.3 / 09.5 / 62.6	23.9 / 09.7 / 13.3 / 63.2	11.6 / 05.5 / 09.1 / 58.4	16.2 / 06.3 / 09.9 / 62.4	29.7 / 13.2 / 17.7 / 64.1
Setting 2: Fine-Tuned Traditional Multi-Lingual Language Models						
mBART-50	27.4 / 11.9 / 19.9 / 67.8	**27.2 / 12.5 / 20.1** / 67.8	26.6 / 12.7 / 20.1 / 65.3	32.9 / 17.5 / 24.9 / **71.0**	25.8 / 11.1 / 19.5 / 68.1	23.2 / 10.8 / 16.7 / 65.4
PISCES	**30.8 / 15.0 / 22.8 / 68.6**	**27.2** / 12.0 / 19.8 / **68.7**	**28.2 / 12.9 / 20.9 / 66.0**	**34.1 / 18.0 / 26.8** / 70.9	**36.3 / 20.7 / 28.8 / 71.9**	**24.3 / 11.0 / 17.4 / 65.7**
Setting 3: Instruction-Tuned LLMs						
LLaMa-2-13B	37.7 / 21.2 / 29.4 / **74.4**	**37.1** / 20.0 / 27.2 / 74.2	40.2 / 25.1 / 32.2 / 74.2	40.3 / 24.3 / 32.4 / 75.4	33.0 / 17.1 / 26.6 / 73.4	38.2 / 20.5 / 26.2 / 73.4
LLaMa-2-7B	35.5 / 19.6 / 27.0 / 73.0	34.9 / 18.4 / 26.5 / 73.2	37.6 / 22.8 / 29.2 / 73.2	37.9 / 22.4 / 30.8 / 75.0	31.8 / 15.8 / 25.9 / 72.6	37.8 / 20.2 / 25.7 / 72.7
LLaMa-3-8B	36.2 / 20.0 / 27.5 / 73.4	35.4 / 18.7 / 26.9 / 73.4	37.9 / 23.5 / 29.8 / 73.9	38.6 / 22.8 / 31.5 / 75.7	32.5 / 16.4 / 26.4 / 73.2	38.5 / 20.7 / 26.2 / 73.3
Vicuna-13B	37.3 / 21.7 / 28.7 / 73.9	36.3 / **20.9** / 28.0 / 73.6	39.8 / 26.2 / 32.3 / **74.4**	40.4 / 24.5 / 32.3 / 75.9	34.2 / 17.6 / 28.1 / 73.5	38.4 / **20.9 / 26.6** / 73.7
Vicuna-13B-16k	**38.0 / 23.0 / 30.2** / 74.1	36.9 / **20.9 / 28.6 / 74.7**	**40.4 / 26.7 / 32.9** / 74.2	**41.2** / 24.6 / 33.6 / 75.9	**34.5** / 18.4 / 28.5 / 73.9	38.3 / 20.6 / 26.4 / 73.5
Vicuna-7B	35.6 / 20.8 / 28.0 / 73.1	34.5 / 19.0 / 26.2 / 73.8	38.3 / 23.9 / 29.1 / 73.3	38.9 / 23.0 / 32.3 / 75.4	31.2 / 15.5 / 25.7 / 72.9	36.8 / 19.4 / 24.5 / 72.4
Vicuna-7B-16k	36.2 / 21.3 / 28.6 / 73.7	35.5 / 20.1 / 27.7 / 73.4	38.7 / 24.2 / 30.3 / 74.2	38.9 / 23.0 / 31.8 / 75.3	32.3 / 15.6 / 25.9 / 72.2	37.7 / 20.8 / 26.5 / 73.3
Baichuan2-13B	36.1 / 20.8 / 28.0 / **74.4**	35.9 / 20.0 / 26.4 / 73.1	38.0 / 24.8 / 29.8 / 73.2	40.7 / 24.8 / 34.1 / 75.9	33.5 / 17.0 / 25.7 / 73.7	39.2 / 20.7 / 25.3 / 73.8
Baichuan2-7B	35.0 / 19.9 / 27.4 / 73.5	35.4 / 19.4 / 26.6 / 73.0	37.3 / 23.0 / 29.4 / 73.3	38.8 / 22.8 / 31.5 / 74.9	31.8 / 14.8 / 23.9 / 72.6	38.1 / 19.9 / 25.9 / 73.7
Qwen-14B	37.1 / 21.9 / 28.4 / 74.2	36.0 / 19.6 / 26.8 / 73.2	38.4 / 24.5 / 30.8 / 73.4	40.2 / 25.4 / 33.4 / 75.5	34.4 / 19.0 / 28.5 / 74.4	39.1 / 20.6 / 26.4 / 74.4
Qwen-7B	34.8 / 18.6 / 26.8 / 73.2	33.5 / 18.3 / 25.0 / 72.5	36.1 / 21.1 / 27.6 / 73.0	38.9 / 22.9 / 31.5 / 75.2	33.1 / 16.1 / 25.7 / 73.2	37.0 / 19.6 / 24.3 / 73.0
Qwen2.5-14B	37.8 / 22.5 / 29.3 / 74.3	36.7 / 20.3 / 28.0 / 74.0	39.4 / 25.6 / 31.6 / 73.9	40.8 / 25.7 / 33.6 / 75.8	34.4 / **19.2 / 28.8 / 74.7**	**39.4** / 20.8 / **26.6 / 74.8**
Qwen2.5-7B	35.2 / 19.0 / 27.3 / 73.7	34.2 / 18.7 / 25.6 / 72.9	36.6 / 21.4 / 28.2 / 73.7	39.5 / 23.6 / 32.3 / 76.0	33.4 / 16.5 / 26.5 / 73.6	37.7 / 20.0 / 24.9 / 73.6
Internlm2-20B	36.7 / 21.5 / 28.2 / 73.7	35.0 / 19.2 / 25.6 / 73.0	38.0 / 24.1 / 29.0 / 72.8	41.1 / **26.0 / 34.2 / 76.2**	**34.5** / 19.0 / 27.9 / 74.3	39.3 / 20.4 / 25.7 / 73.4
Internlm2-7B	35.7 / 19.9 / 27.2 / 73.5	34.1 / 18.7 / 25.6 / 72.6	36.2 / 22.0 / 28.4 / 72.8	40.7 / 25.4 / 33.5 / 75.9	33.3 / 17.7 / 27.0 / 73.3	37.8 / 20.6 / 25.6 / 73.0

Table 11: Experimental results of the overall performance and fine-grained results in each domain. The **bold** denotes the best performance under each setting. Encyc.: Encyclopedia; Tech.: Technology.

on the decoder side and serves as the decoder start token to control which language should be generated in the summaries. Besides, we set the maximum number of tokens for input sequences to 1024 (mBART-50 and PISCES accept input text with a maximum length of 1K, and this is also a shortcoming of traditional models compared with LLMs). The fine-tuned traditional models use the same decoding strategy as the LLMs.

Training/Tuning Hours. All experiments are conducted on NVIDIA A800 GPUs with 80G memory, and we use its GPU hours to denote the consumption of computing resources. Each instruction-tuned 7B LLM needs 19 GPU hours, while each 13B LLM needs 32 GPU hours. For Internlm2-chat-20B, it costs 80 GPU hours since we offload the optimizer. To fine-tune the traditional multi-lingual language models, 3 GPU hours are cost.

F Full Results

Table 11 shows the full results in terms of R1, R2, RL and BS. Typically, the results in terms of R2 are consistent with the results in terms of other metrics.

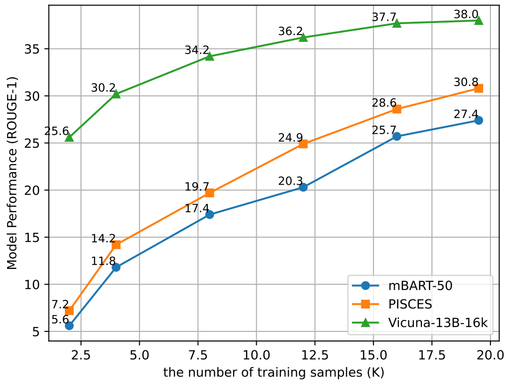

Figure 5: Model performance (ROUGE-1) using different scales of training samples.

G The Effects of Training Scales

To assess the impact of training scales, we randomly select the training samples into various sizes (2K, 4K, 8K, 12K, 16K, and 19.5K) and subsequently fine-tune the models mBART-50, PISCES, and Vicuna-13B-16k for each scale. During the randomly selection, we use probability sampling to ensure the balance of each language as well as each domain. As shown in Figure 5, compared with LLMs, the performance of traditional models is more sensitive with the training scale. Specifically, when decreasing the training data from 19.5K to 2K, mBART-50 and PISCES sacrifice 21.8 and 23.6 R1 (ROUGE-1) scores, respectively, while the counterpart of Vicuna-13B-16k is 12.4.

H Human Evaluation

Following Gao et al. (2023), we employ three graduate students with high levels of fluency in both English and Chinese as our evaluators. We randomly select 100 English documents from the testing set, and let the evaluators judge whether factual errors in the Chinese summaries generated by GPT-4, zero-shot & tuned LLaMa-2-13B, and zero-shot & tuned Vicuna-13B-16k. If a generated summary has factual errors, evaluators also should label which types of factual errors in the summary. Finally, the Fleiss' Kappa scores (Fleiss, 1971) of hallucination error, particulars error, predicate error and entity error are 0.78, 0.73, 0.81, 0.75, respectively, indicating a good inter-agreement among our evaluators.