
Advancing Dialogue Systems: Synthetic Dialogue Generation and Code-Switching in LLMs



Thirunavukarasu Suresh Sathya Krishnan

College of Computing & Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Master of Science, Artificial Intelligence

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

April 2025

.....

Date

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU

Sathya Krishnan

Thirunavukarasu Suresh Sathya Krishnan

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

April 2025

.....

Date



.....

Prof. Chng Eng Siong

Authorship Attribution Statement

This thesis contains material from 1 paper accepted at a conference in which I am listed as an author.

Chapter 3 is published as [Sathya Krishnan Suresh, Wu Mengjun, Tushar Pranav, and Chng Eng Siong. DiaSynth: Synthetic Dialogue Generation Framework for Low Resource Dialogue Applications. Findings of the Association for Computational Linguistics: NAACL 2025](#)

The contributions of the co-authors are as follows:

- Prof. Chng Eng Siong provided the initial research direction
- I proposed the key research idea and implemented the experiments.
- Wu Mengjun and Tushar Pranav helped with additional experiments and result analysis

April 2025

.....

Date

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU

Sathya Krishnan

Thirunavukarasu Suresh Sathya Krishnan

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me throughout the completion of this thesis. First, I am thankful to my supervisor, Prof. Chng Eng Siong, for his support and guidance.

I would also like to extend my appreciation for the support from the NTU Speech Lab. I am grateful for the guidance from Mr Lim Zhi Hao and Mr.Tanmay Surana and the discussions with Mr.Wu Mengjun. The guidance and discussions have been instrumental in shaping the outcomes of this thesis.

Lastly, I would like to thank my family and friends for their support.

Abstract

This thesis is focussed on development of a synthetic dialogue generation system and the evaluation of large language models' (LLMs) understanding of code-switching.

Dialogue systems, used in applications like virtual assistants and customer support, need domain-specific datasets to work well. Without these datasets, they struggle to interact effectively. People from bilingual or multilingual backgrounds often switch between languages, a behavior known as code-switching. The system's ability to understand this impacts its effectiveness.

This thesis introduces DiaSynth, a synthetic dialogue generation framework, and CS-Sum, a benchmark for code-switching dialogue summarization. DiaSynth uses an LLM to generate realistic conversations in domains with limited data by incorporating personas and Chain of Thought reasoning. CS-Sum, with three code-switching language pairs, highlights the limitations of LLMs in understanding code-switched dialogues.

Experiments show that DiaSynth improves the performance of the LLMs on downstream tasks by 16.47% and it is able to replicate performance of models trained on human generated data up to 90.48%. Testing 10 LLMs on CS-Sum reveals that state-of-the-art models struggle more with code-switching than with English, and the thesis examines the specific errors they make.

Contents

Acknowledgements	ix
Abstract	xi
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Motivation	3
1.2 Contributions	4
1.3 Outline of the Thesis	5
2 Literature Review	7
2.1 Synthetic Dialogue Generation	8
2.1.1 Personality in Synthetic Data Generation	8
2.1.2 Prompting Task-Oriented Dialogue Generation	8
2.1.3 Existing Task-Oriented Dialogue Datasets	9
2.2 Code-Switching	9
2.2.1 Current State	9
2.2.2 Code Switching Datasets	10
LinCE:	10
GLUECoS:	10
GupShup:	11
ASCEND:	11
CroCoSum:	11
2.3 Summary	12
3 DiaSynth: Synthetic Dialogue Generation Framework for Low Resource Dialogue Applications	13
3.1 DiaSynth	13
3.1.1 Subtopic Generation	15
3.1.2 Persona Generation	15
3.1.3 Characteristics	16

3.1.4	Dialogue Generation	16
3.2	Experimental Setup	17
3.2.1	Quality of the dialogues	17
3.2.1.1	FED	17
3.2.1.2	GPTScore	18
3.2.1.3	G-Eval	19
3.2.1.4	Evaluation Process	20
3.2.2	Downstream Task - Summarization	20
3.2.2.1	QAGS	20
3.2.2.2	BERTScore	21
3.2.2.3	Evaluation Process	22
3.3	Results	22
3.3.1	Quality of the Dialogues	23
3.3.1.1	Metric Scores	23
3.3.1.2	Strong performance of LLaMA-3	25
3.3.2	Fine-tuning and Performance Results	25
3.3.3	Additional Downstream Task: Response Generation	28
3.4	Ablations and Hallucination Study	29
3.4.1	Ablation study	29
3.4.2	Hallucination study	31
3.4.2.1	SelfCheckGPT	31
3.4.2.2	ChainPoll	31
3.4.2.3	Implications for DiaSynth	32
3.5	Summary	32
4	CS-Sum: A Benchmark for Code-Switching Dialogue Summarization and the Limits of Large Language Models	33
4.1	CS-Sum Benchmark	34
4.1.1	CS Metrics	34
4.1.2	Gemini generated training data	35
4.2	LLM performance on CS-Sum	37
4.2.1	Evaluation Metrics	38
4.2.1.1	SBERT-Cosine	38
4.2.1.2	Jaccard	39
4.2.1.3	Meteor	40
4.2.2	Few-shot and Translate-Summarize	41
	Few-shot	41
	Translate-Summarize	41
4.2.3	LoRA and QLoRA Fine-Tuning for CS Dialogue Summarization	41
4.2.4	Result Analysis	42
	Translate-Summarize % Improvement	44
	LoRA and QLoRA Results	45

4.2.5	Training on filtered data	47
4.3	Error analysis	48
4.4	Summary	51
5	Conclusion and Future Work	55
5.1	Conclusion	55
5.2	Future Work	56
	Bibliography	59

List of Figures

3.1	DiaSynth Architecture	14
4.2	Wrong summary with high BERTScore 0.903	45
4.3	Distribution b/w filtered CS-Sum-Syn and CS-Sum for EN-ZH	47
4.4	Filtered data % improvement over CS-Sum-Syn for EN-ZH	49
4.5	CSL error example	50

List of Tables

3.1	Characteristics of the Dialogue for CoT Prompt	16
3.2	Data Statistics	23
3.3	FED scores	24
3.4	GPTScore	24
3.5	G-EVAL	25
3.6	Performance of models before and after finetuning on in-domain data	26
3.7	Performance after finetuning on synthetic data	27
3.8	Summarization results for Dialoguesum and SAMSum	28
3.9	BERTScore evaluation of response generation models fine-tuned on synthetic and in-domain datasets.	28
3.10	FED scores with DialogSum and SAMSum as base	30
3.11	GPTScore with DialogSum and SAMSum as base	30
3.12	Hallucination calculation for DialogSum and SAMSum few-shot data	32
4.1	CS metrics measured on the CS-Sum dataset	35
4.2	Divergence	37
4.3	Hyperparameters used for LoRA and QLoRA fine-tuning.	42
4.4	Few-Shot Performance/Translate-Summarize % improvement on CS- Sum	43
4.5	LoRA / QLoRA % Improvement over Few-Shot on CS-Sum (EN-TA)	46
4.6	Comparison of error rates (%) in Few-Shot vs. LoRA fine-tuned models across different CS language pairs.	50
4.7	Error examples	52
4.8	Error examples	53
4.9	Error examples	54

Chapter 1

Introduction

This thesis addresses two critical challenges in dialogue systems: (1) generating high-quality synthetic dialogues for specialized English domains with limited existing datasets, and (2) evaluating Large Language Models’ (LLMs) comprehension and summarization capabilities in multilingual, code-switched contexts. Specifically, it introduces DiaSynth (Chapter 3), a synthetic dialogue generation framework for English-domain dialogues, and CS-Sum (Chapter 4), the first benchmark designed for assessing summarization of code-switched dialogues.

Dialogue systems are essential in natural language processing (NLP), with applications spanning customer service chatbots, virtual assistants, and healthcare bots. These systems rely heavily on large, high-quality, domain-specific datasets to effectively simulate human-like conversations. However, obtaining such datasets is often a significant challenge, especially for specialized domains like academic discussions, healthcare, and everyday conversations. The difficulty in acquiring domain-specific data arises due to the high costs and complexity of data collection, which is further increased when working with low-resource languages or niche topics. As a result, dialogue systems often struggle to generalize across multiple domains, limiting their ability to provide effective and scalable solutions in real-world applications [1–3].

Existing approaches have attempted to address this issue by collecting domain-specific dialogue datasets; however, these efforts face significant limitations. On one hand, some datasets are shallow, capturing only basic conversational exchanges that lack the depth and specificity required for domain expertise. On the other hand, niche domain datasets, while contextually rich, suffer from limited scale,

making it challenging to develop dialogue systems that can handle a wide range of real-world scenarios. This imbalance between depth and scale severely hampers the development of dialogue systems in underrepresented domains, where data collection is not only costly but also logistically complex. As such, there remains a pressing need for scalable, high-quality data generation methods that can overcome these limitations and provide the necessary diversity and richness for robust dialogue system development [4, 5].

In addition to the challenges posed by data scarcity, another critical issue in dialogue systems is the difficulty in generating realistic and varied conversations. To produce realistic dialogues, researchers have increasingly turned to persona-based generation methods, where synthetic dialogues are enriched by simulating diverse personalities and characteristics. This approach has been shown to improve the realism and effectiveness of generated dialogues [4]. However, even with persona-based generation, the scope of conversation and the variety of topics covered often remain limited, especially in specialized domains. Thus, while personality-driven dialogue generation has its benefits, it still faces challenges in terms of scalability and domain-specific richness, which are crucial for creating effective dialogue systems. Thus, frameworks such as DiaSynth (Chapter 3) directly tackle the foundational challenge of dialogue data scarcity, enabling training and deployment of dialogue systems in specialized English domains.

However, effective real-world deployment of dialogue systems goes beyond generating English dialogues. It also demands strong evaluation of these systems' performance in multilingual contexts, especially where code-switching is prevalent. Code-switching (CS), the practice of alternating between two or more languages in a single conversation, is a prevalent phenomenon in multilingual societies and has become common in social media interactions. Bilingual and multilingual speakers often engage in CS, which adds a layer of complexity to language processing tasks. Despite the widespread occurrence of CS, current NLP models, particularly LLMs, struggle to fully comprehend and process CS dialogues [6], [7], [8]. Most LLMs are predominantly trained on monolingual data, with a heavy focus on English, making it challenging for them to handle mixed-language input effectively. As a result, the performance of LLMs is hindered when faced with CS, leading to difficulties in accurately interpreting multilingual prompts and producing coherent outputs.

Existing benchmarks for evaluating the performance of LLMs in multilingual settings, such as LinCE [9] and GLUECoS [10], primarily focus on word-level tasks like language identification (LID), named entity recognition (NER), and part-of-speech (POS) tagging. While these benchmarks are useful for assessing certain aspects of multilingual understanding, they fall short of addressing the complexities of CS dialogue, which requires deeper discourse-level comprehension. Summarization, a task that demands both an understanding of the core content and the ability to produce a coherent summary, presents an ideal challenge for evaluating LLMs' CS capabilities. However, there is currently a lack of datasets specifically designed for CS dialogue summarization, with the only existing dataset, Gupshup [11], limited to a single language pair (English-Hindi). This gap in CS-specific benchmarks highlights the need for more comprehensive tools to assess how well LLMs can handle the intricacies of code-switched conversations.

1.1 Motivation

The development of strong dialogue systems relies heavily on the availability of large, high-quality, domain-specific datasets. However, acquiring such datasets is often challenging, especially for specialized domains like healthcare, academic discussions, and everyday conversations. Existing dialogue datasets typically face two key problems: either they are shallow, lacking in depth and domain-specific detail, or they are contextually rich but suffer from limited scale. This imbalance hampers the development of dialogue systems that can generalize well across different topics and domains, creating significant barriers to building scalable, adaptable systems.

Similarly, handling CS data in LLMs remains a complex challenge. Code-switching, the practice of alternating between two or more languages in a single conversation, is common in multilingual societies and on social media. However, LLMs, which are primarily trained on monolingual datasets, struggle to process and understand these mixed-language dialogues effectively. Current benchmarks for evaluating CS understanding often focus on word-level tasks like language identification or named entity recognition, which fail to assess the more complex discourse-level understanding required for tasks such as dialogue summarization. This leaves a gap in evaluating how well LLMs can handle the intricacies of real-world multilingual communication.

This thesis aims to address these challenges by introducing solutions that improve the data available for dialogue systems and provide a strong benchmark for evaluating CS comprehension. Specifically, DiaSynth is proposed as a synthetic dialogue generation framework that enables the creation of high-quality, domain-specific dialogues, overcoming the data scarcity issue in dialogue system development. In addition, the CS-Sum benchmark is introduced to evaluate LLMs on the more complex task of CS dialogue summarization, providing a comprehensive tool for assessing the CS comprehension capabilities of current models. By addressing these problems, this thesis contributes to powering future research in both dialogue system development and multilingual NLP.

1.2 Contributions

This thesis presents two significant contributions in the field of natural language processing. First, we introduce **DiaSynth**, a synthetic dialogue generation framework that addresses the challenge of data scarcity for dialogue systems. The contributions of DiaSynth are as follows:

- **DiaSynth Framework:** We propose a scalable framework for generating high-quality, domain-specific dialogues using LLMs and CoT reasoning, which simulates diverse and realistic dialogues tailored to different domains.
- **Evaluation on Downstream Tasks:** We demonstrate the effectiveness of DiaSynth by evaluating its generated data on dialogue summarization tasks. Models fine-tuned on DiaSynth data show an average performance improvement of 16.47
- **Flexibility Across LLMs:** DiaSynth’s ability to generate high-quality data using various LLMs, including both open-source and proprietary models, makes it adaptable and robust for training dialogue systems in different domains.

Secondly, this thesis introduces CS-Sum, a benchmark for evaluating CS dialogue summarization. The contributions of CS-Sum are as follows:

- **CS-Sum Benchmark:** We present the first-ever benchmark for CS dialogue summarization, containing 1.4K CS dialogue-summary pairs across three language pairs—Mandarin-English, Tamil-English, and Malay-English—enabling comprehensive evaluation of LLMs on this complex task.
- **Evaluation of LLM Performance:** We conduct a detailed evaluation of 10 state-of-the-art LLMs on CS-Sum, using multiple evaluation settings like few-shot and fine-tuning, and provide in-depth analysis of model performance, highlighting their struggles with code-switching dialogue comprehension.
- **Error Analysis and Insights:** We perform a thorough error analysis, categorizing the common errors made by LLMs in summarizing CS dialogues, including Code-Switching Loss (CSL), Meaning Shift from Poor Translation (MST), and Speaker Misattribution (SMA).

1.3 Outline of the Thesis

This thesis is divided into 5 chapters. The following is an overview of each chapter:

- Chapter 2 reviews relevant content on synthetic dialogue generation and various benchmarks available for evaluating NLP models on code-switching tasks
- Chapter 3 discusses DiaSynth, the proposed framework for generating realistic and large scale synthetic dialogues.
- Chapter 4 introduces the CS-Sum benchmark, along with a detailed analysis on the struggles of LLMs when it comes to CS.
- Chapter 5 concludes the thesis and discusses the future work.

Chapter 2

Literature Review

This chapter presents a comprehensive review on synthetic dialogue generation and the current state of code-switching NLP.

In the current world, dialogues systems are everywhere, powered by deep learning models and they are present in almost all fields, ranging from healthcare to customer support. For the models to be really effective they need to be trained on high quality dialogue datasets that would enable the models to interact effectively and accurately with the users. Since acquiring domain specific dialogues is too costly and existing datasets are either too generic [12, 13] or small in scale [2], synthetic data with personas is seen as an effective solution to address the problems of scale and domain specificity. Section 2.1 provides a comprehensive review on current state of synthetic dialogue generation.

The study of CS in NLP has gained significant traction in recent years, since speakers who speak more than one language are far more than monolingual speakers [14] and recent studies [15] have shown that people prefer to interact with LLMs in code-switch. Code-switching [16], where speakers alternate between languages within a single conversation or even sentence, has traditionally been explored using linguistic theories. However, with the rise of multilingual LLMs, researchers have increasingly turned to machine learning-based approaches to better handle CS in real-world applications. Despite this progress, LLMs still face notable challenges in effectively processing CS data, primarily due to issues related to data scarcity, model robustness, and task-specific evaluation. Section 2.2 provides a comprehensive review on the current state of Code-Switching NLP.

2.1 Synthetic Dialogue Generation

2.1.1 Personality in Synthetic Data Generation

In recent years, there has been a significant increase in research focused on synthetic dialogue generation, largely driven by advancements in LLMs. To generate **realistic and diverse** synthetic data, researchers have incorporated personalities, profiles, and character information when prompting LLMs to generate dialogues [4]. By enhancing dialogue realism through the simulation of various personality profiles, utilizing the Big Five personality model, and employing structured prompts, this approach has improved task performance in models fine-tuned on these generated dialogues compared to those trained on general chit-chat datasets.

Moreover, integrating personas into synthetic data generation prompts [5] has demonstrated that models fine-tuned on personalized synthetic data outperform some LLMs of much larger scales. The inclusion of personas in prompts provides diversity in difficulty levels and ranges within the synthetic data, enabling the models to handle situations of varying complexity.

2.1.2 Prompting Task-Oriented Dialogue Generation

Prompt-based techniques have also emerged as powerful methods for generating high-quality synthetic dialogues, particularly for task-oriented dialogue systems. [17] explore the generation of synthetic dialogues from structured prompts, focusing on enhancing task-oriented dialogue systems. Their work demonstrates that prompt engineering can produce dialogues that are contextually appropriate and improve system performance by aligning synthetic data more closely with real-world requirements.

To achieve a higher quantity, diversity, and creativity in human-written instruction data, [18] propose inputting prompts to LLMs to generate instructions based on a small set of seed human-written instructions. This approach aligns the expanded training data more closely with desired task objectives and allows for iterative improvements, producing more nuanced and effective dialogues that meet specific task demands.

2.1.3 Existing Task-Oriented Dialogue Datasets

In addition to prompt-based synthetic data generation, various large-scale dialogue datasets have been instrumental in advancing task-oriented dialogue systems. Among these, the MultiWOZ dataset [3] is a prominent resource, providing richly annotated dialogues across multiple domains. MultiWOZ has enabled researchers to train models capable of handling complex, multi-turn interactions across diverse tasks. The nature of MultiWOZ’s annotations has made it a benchmark for evaluating the performance of dialogue systems, though it is often complemented by synthetic data to introduce further diversity and variation in dialogue scenarios.

Similarly, Doc2Dial [1] is another widely used dataset designed specifically for document-grounded dialogue systems. Doc2Dial includes conversations grounded in structured documents, focusing on providing users with accurate and relevant information based on their inquiries. This dataset has been instrumental in improving the ability of dialogue systems to retrieve and generate accurate responses when interacting with complex information sources. However, much like MultiWOZ, Doc2Dial’s scope is limited to the predefined topics and domains covered within the dataset, which can restrict model generalizability to new or unseen situations.

2.2 Code-Switching

2.2.1 Current State

Recent studies [7, 8] have demonstrated that while LLMs like GPT-4 show promise in certain CSW tasks such as sentiment analysis and machine translation, their performance remains inconsistent, particularly when dealing with complex and heavily mixed-language text. These models often struggle with the syntactic and semantic nuances required for handling code-switched dialogue, especially in scenarios with multiple languages or dialects. The lack of extensive, high-quality, and publicly available datasets further exacerbates these challenges, limiting the ability to fine-tune LLMs effectively for CS tasks.

Furthermore, the predominant use of high-resource language pairs, such as English-Spanish or Hindi-English, has led to a lack of diversity in the datasets used to train these models. This bias toward certain language combinations has hindered progress in CSW research for underrepresented languages, such as Tamil-English or Arabic-English. Researchers have called for a more inclusive approach to CS research, focusing on a wider variety of language pairs and dialects to better reflect real-world multilingual communication [7]. In addition, while neural networks and pre-trained models have shown some promise in improving CSW performance, the lack of interpretability in these models and the ongoing scaling challenges present significant obstacles to their widespread adoption in practical NLP applications [6].

Ultimately, while advancements have been made, there is still a need for more diverse and accessible datasets, as well as improved evaluation metrics, to assess the effectiveness of LLMs in processing code-switched text. The future of CS research lies in scaling models to handle a broader range of languages, enhancing the interpretability of these models, and addressing the methodological gaps that currently limit their performance [19].

2.2.2 Code Switching Datasets

LinCE: The LinCE dataset [9] consolidates multiple corpora to address linguistic code-switching (CS) in NLP. It includes data from four language pairs: Spanish-English, Nepali-English, Hindi-English, and Modern Standard Arabic-Egyptian Arabic. The dataset supports four core tasks: language identification (LID), named entity recognition (NER), part-of-speech (POS) tagging, and sentiment analysis (SA), offering a comprehensive testbed for CS models. With 10 datasets, including 4 for LID, 3 for NER, 2 for POS, and 1 for SA, LinCE contains tens of thousands of tweets. However, it has limitations, including uneven class distribution and challenges in handling low-resource languages.

GLUECoS: GLUECoS [10] is a benchmark for code-switched languages, focusing on English-Hindi and English-Spanish pairs. It covers tasks like LID, POS tagging, NER, sentiment analysis, question answering (QA), and natural language inference (NLI). The dataset includes annotated corpora from various domains, with both word-level and sentence-level tasks. While the dataset is comprehensive,

with large annotations for tasks like NER and sentiment analysis, it faces challenges like the scarcity of high-quality code-switched data and the complexity of code-switching in certain tasks.

GupShup: GupShup [11] is a dataset designed for abstractive summarization of Hindi-English code-switched conversations. It contains over 6,800 multi-party conversations, totaling 76,330 utterances, with corresponding human-annotated summaries in both English and Hindi-English. The primary task is generating English summaries from code-switched conversations, aiding the study of models for both translation and summarization. However, GupShup has limitations, including the complexity of code-switching patterns, informal language, and the relatively small and specialized conversational context, limiting its applicability to diverse real-world scenarios.

ASCEND: ASCEND [20] is a corpus for studying spontaneous Chinese-English code-switching in multi-turn conversations. It includes 10.62 hours of audio from 23 bilingual speakers across Hong Kong, Taiwan, and Mainland China, with 49 informal conversation sessions on topics like education and sports. The dataset features both intra- and inter-sentential code-switching, comprising around 12,300 utterances and 145,146 tokens. ASCEND is primarily used for automatic speech recognition (ASR) model training, with baseline results showing a character error rate (CER) of 22.69%. However, its limitations include a small speaker set and a narrow range of conversational styles, limiting its broader applicability.

CroCoSum: CroCoSum [21] focuses on cross-lingual code-switched summarization, specifically for technology news. The dataset includes over 24,000 English articles and 18,000 human-written Chinese summaries, with 92% of the summaries containing code-switched phrases. The main task is generating Chinese summaries from English articles, with a focus on code-switching between the languages. Although CroCoSum is valuable for studying code-switching in multilingual models, it is limited by its domain-specific focus on technology news and the use of code-switching mainly in named entities, rather than more complex linguistic patterns.

Though other datasets exist [22–24], these datasets were the most relevant due to their recent development and focus on real-world code-switching scenarios. While

current LLMs have become proficient in word-level tasks such as language identification and named entity recognition, these datasets still fall short of addressing the complexity of sentence-level tasks, such as summarization, sentiment analysis, and cross-lingual understanding. This gap highlights the need for a more comprehensive benchmark that highlights the hidden deficiencies in LLMs when handling multiple CS language pairs.

2.3 Summary

This chapter reviewed the current advancements in synthetic dialogue generation and the state of code-switching in NLP. From the review, we conclude that while significant progress has been made, there is still a lack of scalable and domain-specific frameworks for synthetic dialogue generation. Existing methods fail to address these challenges adequately, especially in terms of scalability and domain specificity.

Regarding code-switching in NLP, it is evident that LLMs, though effective at word-level tasks such as LID and NER, still struggle with more complex sentence-level tasks like sentiment analysis or QA. Furthermore, the current code-switching datasets either lack sufficient linguistic diversity (in terms of code-switching language pairs) or fail to tackle sentence-level tasks, leaving a gap in our understanding of the deficiencies in LLMs when handling code-switched data.

Chapter 3

DiaSynth: Synthetic Dialogue Generation Framework for Low Resource Dialogue Applications

In this chapter, we propose DiaSynth - a synthetic dialogue generation framework with a specific focus on scalable and controllable dialogue generation for low resource applications. DiaSynth addresses the problem of lack of large-scale domain specific dialogue datasets through a multi stage process. The steps involved in this multi stage process are generating sub-topics, generating personas and their combinations, using CoT [25] to let the LLM decide the setting of a dialogue and finally generating the dialogue. This chapter is organised as follows: Section 3.1 describes the DiaSynth framework, Section 3.2 describes the experimental setup for using and evaluating DiaSynth, Section 3.3 provides the results, Section 3.4 provides ablation and hallucination studies and finally Section 3.5 provides a summary of the chapter.

3.1 DiaSynth

DiaSynth is a synthetic dialogue generation framework designed to address the scarcity of high-quality, large-scale, domain-specific dialogue datasets. DiaSynth uses an LLM and CoT reasoning to simulate diverse, nuanced dialogues.

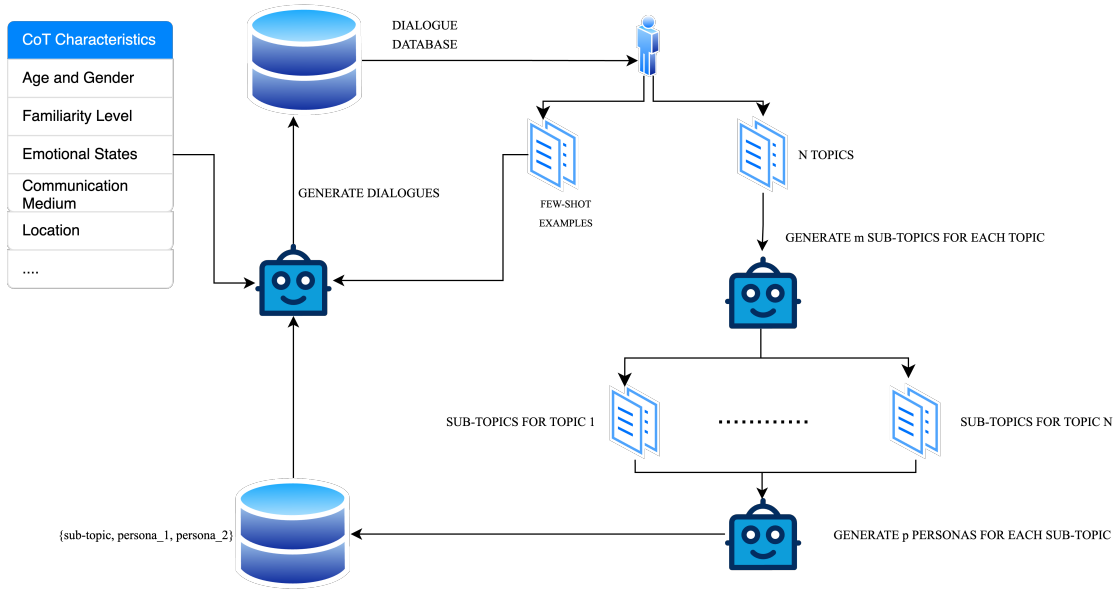


FIGURE 3.1: DiaSynth Architecture

DiaSynth takes a list of user-provided topics to generate dialogues. The users can optionally provide few-shot examples of the format in which they want the dialogue to be generated. Directly generating dialogues from user topics would be too superficial due to their lack of specificity. To overcome this lack of specificity, we generate m sub topics for each of the n topics given by the user. Generating dialogues from the subtopics will have specificity but the dialogues will lack variety. This is because every dialogue is influenced implicitly by the personas of the people involved in the dialogue and, other characteristics such as the location, emotion and more. To enhance variety and depth, we generate p personas per subtopic and create dialogues for all persona-subtopic combinations. To further ground the dialogues in various settings and characteristics, we employ CoT reasoning during the generation process. DiaSynth employs CoT to reason about the settings and characteristics of a dialogue, which are listed in subsection 3.1.3, ensuring that the dialogues are contextually rich and realistic. This three-stage pipeline not only guarantees the quality of the generated dialogues but also allows for exponential scalability.

3.1.1 Subtopic Generation

Subtopic generation is a crucial step in DiaSynth’s pipeline, since it enhances the specificity and depth of the dialogues that will be generated later. For each primary topic given by the user, DiaSynth generates multiple subtopics, effectively narrowing the focus of the conversation. This breakdown is necessary because the primary topics are often too general to generate contextually rich dialogues on their own. For instance, a topic like “healthcare” can be expanded into subtopics such as “doctor-patient consultations,” “mental health discussions,” and “medical diagnostics,” each of which offers a more focused context for dialogue generation. To achieve this, DiaSynth prompts an LLM to generate the user specified number of subtopics for each primary topic. We also run a similarity check between each of the subtopics and remove subtopics that are too similar to other subtopics using a threshold.

3.1.2 Persona Generation

Personas of the individuals involved in a conversation are primary influencers in determining how a conversation pans out. Using random personas from persona datasets and prompting the LLM to simulate a dialogue between them about a random topic often leads to superficial dialogues that lack depth and contextual richness. To address this issue, DiaSynth generates a user-specified number of personas for each subtopic, ensuring that the personas are conditioned on the subtopic context. This conditioning prompts the LLM to create personas that are most likely to engage in a meaningful dialogue about the subtopic, such as a medical professional and a patient discussing “medical diagnostics” or a researcher and a student talking about “academic publishing.” We also run a similarity check for the personas too. The conditioned persona generation is crucial because it ensures that future dialogues will not only be contextually rich but also exhibit a high level of depth. Each dialogue will be between two personas who have relevant expertise or perspectives on the given subtopic, allowing the conversation to explore nuances that would otherwise be missed in a generic dialogue setting. We present the impacts of sub-topics and personas in [Section 3.4](#)

3.1.3 Characteristics

Table 3.1 shows different characteristics that we let the LLMs reason and decide using CoT. Before generating the dialogues, the LLMs are prompted to first reason about the various characteristics list for the dialogue given the topic and the personas.

Characteristic	Description
Age and Gender	Defines demographic details, influencing style and tone.
Familiarity Level	Affects formality and depth based on relationship between speakers.
Emotional States	Impacts tone and flow based on emotions (e.g., happy, sad).
Formality Level	Determines level of politeness or casualness.
Duration of the Conversation	Suggests the intended length and complexity of dialogue.
Communication Medium	Defines the medium, influencing style.
Topic of the Conversation	Guides the content and direction of the dialogue.
Location of the Conversation	Adds context influencing formality and content.
Agreement or Disagreement	Drives dialogue dynamics based on agreement level.
Natural Dialogue Features	Adds authenticity with fillers, pauses, and slang.

TABLE 3.1: Characteristics of the Dialogue for CoT Prompt

3.1.4 Dialogue Generation

The final stage in DiaSynth’s pipeline is the generation of dialogues, where all the components—subtopics, personas, and characteristics—converge to create contextually rich and realistic conversations. This step uses an LLM as the backbone and CoT as the reasoning mechanism, allowing the model to simulate dialogues that incorporate various aspects of human interaction. DiaSynth generates dialogues by pairing all persona-subtopic combinations. The process also integrates predefined characteristics (Table 3.1) like emotional state, formality, and familiarity to guide the flow and style. These characteristics are defined in the CoT prompt, guiding the LLM to generate realistic, contextually appropriate dialogues. The

importance of CoT and the lack of it affects the quality of the dialogues, which is shown quantitatively in Section 3.4.

3.2 Experimental Setup

In this section, we detail the experimental setup used to evaluate the effectiveness of DiaSynth. Our evaluation focuses on two criteria - quality of the dialogues generated and usability of the dialogues generated for a downstream task. Quality of the dialogues is evaluated using metrics such as FED, GPTScore, and G-Eval. We evaluate the usability of DiaSynth-generated dialogues by using summarization as the downstream task.

3.2.1 Quality of the dialogues

To evaluate the quality of the dialogues, we employ the metrics that have been developed for evaluating the quality of text generated by LLMs. We use the FED [26], GPTScore [27] and G-Eval [28] metrics which are described in the following subsections.

3.2.1.1 FED

The FED (Fine-grained Evaluation of Dialogue) metric is an unsupervised and reference-free automatic evaluation metric for dialogue quality. It uses a pre-trained model, DialoGPT [29], to estimate the quality of a system’s responses by calculating the likelihood that a follow-up response (such as "That’s interesting!") would be generated by DialoGPT. The key idea behind FED is to measure the likelihood of various follow-up utterances in response to a system’s output, without relying on any reference response. This allows it to evaluate a wide range of fine-grained qualities of the dialogue, both at the turn level and across the entire conversation.

The formula for the FED score for a given response r in a dialogue context c is computed as follows:

$$\text{FED}(c, r) = \sum_{i=1}^{|p|} D(c + r, p_i) - \sum_{i=1}^{|n|} D(c + r, n_i)$$

Where:

- $D(c + r, p_i)$ is the log-likelihood of DialoGPT generating a positive follow-up utterance p_i after the context c and the system response r .
- $D(c + r, n_i)$ is the log-likelihood of generating a negative follow-up utterance n_i after the context c and response r .

The follow-up utterances for each quality include a set of positive and negative examples, which are hand-written and minimally tuned on a small subset of the data. These are designed to reflect various dialogue qualities, such as being interesting, engaging, fluent, relevant, etc.

3.2.1.2 GPTScore

GPTScore is an evaluation framework designed to score generated texts by using the emergent abilities of GPTs. Unlike traditional evaluation methods that rely on model fine-tuning, GPTScore operates in a training-free manner, enabling multi-faceted evaluation across various aspects. This framework works by assigning higher probabilities to higher-quality generated texts, based on specific task specifications and aspect definitions. By utilizing large pre-trained models such as GPT-3, GPTScore evaluates the likelihood of text generation under the given context, allowing for a customizable, aspect-specific assessment.

The core of GPTScore is the conditional probability defined as follows:

$$\text{GPTScore}(h|d, a, S) = \sum_{t=1}^m w_t \log p(h_t|h_{<t}, T(d, a, S), \theta)$$

where $h = \{h_1, h_2, \dots, h_m\}$ represents the generated text, d is the task description, a is the aspect definition, and S is the context (e.g., source text or reference). $T(d, a, S)$ is the prompt template, and w_t is the weight of the token at position t . The evaluation protocol is set through manual prompt engineering, which defines

how the task description, aspect, and context are organized. For evaluating text quality in terms of specific aspects, GPTScore uses evaluation aspects such as coherence, relevance, fluency, and informativeness.

3.2.1.3 G-Eval

G-Eval is a reference-free framework designed for evaluating the quality of text generated by Natural Language Generation (NLG) systems, using LLMs such as GPT-4. It works by employing a CoT approach along with a form-filling paradigm to assess the quality of NLG outputs. The system utilizes LLMs to generate a detailed evaluation step-by-step, which is then used to evaluate various aspects of the generated text, such as coherence, relevance, and fluency.

The evaluation process follows a structured methodology:

- **Prompt-based Evaluation:** The LLM is prompted with a task introduction and evaluation criteria, which define the context and metrics for evaluation.
- **Chain-of-Thought (CoT):** The LLM generates intermediate steps (CoT) that help break down the evaluation process into detailed steps. This allows for a more nuanced and transparent evaluation of the text.
- **Form-Filling Scoring:** The evaluator then scores the generated text based on the evaluation steps, and the final score is obtained through a weighted summation of the probabilities of different rating tokens output by the LLM.

The formula for the final score is computed as:

$$\text{score} = \sum_{i=1}^n p(s_i) \times s_i$$

where $p(s_i)$ is the probability of each score s_i (from a predefined set of scores) and the final score is a weighted sum based on these probabilities.

The framework is designed to be flexible and adaptable, providing a robust method for evaluating the quality of generated text without relying on human-generated references.

3.2.1.4 Evaluation Process

To validate the framework across models and also domains, we generate data using three open source LLMs, one closed source LLM and also use few shot examples from two different dialogue datasets. The open sourced LLMs are - **Phi-3** Abdin et al. [30], **InternLM-2.5** Cai et al. [31], **LLaMA-3** Dubey et al. [32] and the closed source LLM used is **GPT-4o**. The 8-bit quantized versions of the open source LLMs were used for faster experimentation and generation. The two different dialogue datasets that were used as few-shot examples are DialogSum Chen et al. [12] and SAMSum Gliwa et al. [13]

3.2.2 Downstream Task - Summarization

To evaluate the usability of the dialogues generated by DiaSynth, we choose summarization as the downstream task. Summarization, a key application of dialogue systems, aims to generate concise, contextually relevant summaries. We use three established evaluation metrics—QAGS [33], BERTScore [34], and ROUGE-L [35]—to assess the performance of summarization models fine-tuned on DiaSynth-generated data. We describe the QAGS and BERTScore metrics in the following subsections.

3.2.2.1 QAGS

QAGS (Question Answering and Generation for Summarization) is an automatic evaluation metric designed to assess the factual consistency of generated summaries. The metric aims to identify factual inconsistencies by asking questions about the summary and comparing the answers with the original source. The underlying intuition of QAGS is that a factually consistent summary should generate similar answers to questions based on both the summary and the source text. This approach overcomes limitations of traditional metrics like ROUGE, which are insensitive to factual errors.

The framework consists of three main components:

- **Question Generation (QG):** A question generation model is used to generate a set of questions based on the summary. These questions are designed to target key pieces of information in the summary.
- **Question Answering (QA):** A question answering model is then used to answer these questions based on both the source text and the generated summary. The answers from both sources are compared.
- **Answer Similarity:** The factual consistency score is calculated by comparing the corresponding answers using a similarity function, such as token-level F1 score.

The factual consistency of a summary is quantified by the following formula:

$$QAGS(Y) = \mathbb{E}_{Q \sim p(Q|Y)} [D(p(A|Q, X), p(A|Q, Y))]$$

Where: Q represents a generated question, $p(A|Q, X)$ and $p(A|Q, Y)$ are the answer distributions given the source article X and the summary Y , respectively. D is a similarity function (e.g., token-level F1 score) that measures how similar the answers are.

In practice, QAGS uses a sequence-to-sequence model for generating questions and an extractive QA model for answering the questions. The system compares the generated answers with the original article answers to identify factual errors. The final score is the average similarity across all questions.

3.2.2.2 BERTScore

BERTScore is a metric for evaluating the quality of generated text by leveraging contextualized embeddings from pre-trained models such as BERT [36]. It measures the similarity between a candidate sentence and a reference sentence by computing the cosine similarity of token-level embeddings. BERTScore addresses two common issues in traditional n-gram-based metrics: it effectively matches paraphrases and handles sentence reordering, thus capturing more nuanced semantic meaning.

The score is computed by matching each token in the reference sentence to the most similar token in the candidate sentence. This is done by calculating the cosine similarity between their contextual embeddings, and then combining precision and recall into an F1 score. In BERTScore, the precision (P_{BERT}), recall (R_{BERT}), and F1 score (F_{BERT}) for a reference sentence x and a candidate sentence \hat{x} are computed as follows:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{x_j \in \hat{x}} x_i^T x_j$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T x_j$$

$$F_{\text{BERT}} = \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

Where x_i and \hat{x}_j represent token embeddings for reference and candidate sentences, respectively.

3.2.2.3 Evaluation Process

We fine-tune pretrained summarization models like DistilBART, BART [37], T5 [38] and LED [39], on DiaSynth-generated dialogues and evaluate their performance using the above metrics. We evaluate the usability of DiaSynth in two key aspects: first, by assessing the performance improvement of models fine-tuned on DiaSynth-generated data compared to the pretrained models; and second, by measuring the extent to which DiaSynth-generated data reflects real-world data distribution by comparing the performance of models fine-tuned on DiaSynth data versus those fine-tuned on in-domain data. We also present the results of finetuning BART and T5 on synthetic data and in-domain data on response generation in subsection 3.3.3.

3.3 Results

This section discusses the results of the data generated using DiaSynth (quality of the data and usability in downstream tasks) with different LLMs and varying

LLM	Few-shot	# Samples	Avg. # of turns	Avg. # of tokens per turn	Diversity
Phi-3	DialogSum	1215	9.13	20.38	0.27
InternLM-2.5	DialogSum	1035	9.23	27.98	0.30
LLaMA-3	DialogSum	1154	6.86	31.99	0.29
GPT-4o	DialogSum	1375	15.16	15.96	0.29
Phi-3	SAMSum	1410	13.98	13.94	0.27
InternLM-2.5	SAMSum	1135	13.96	19.07	0.29
LLaMA-3	SAMSum	1195	10.54	20.41	0.29
GPT-4o	SAMSum	1380	15.43	13.53	0.28

TABLE 3.2: Data Statistics

few-shot examples. Specifically, we utilized Phi-3, InternLM-2.5, LLaMA-3 and GPT-4o as the LLM backbones, and the few-shot examples were sourced from DialogSum and SAMSum datasets. These combinations allow us to evaluate the robustness and adaptability of DiaSynth across different models and few shot examples. In total, eight distinct datasets were generated using DiaSynth by pairing each LLM with the two sets of few-shot examples, resulting in all possible combinations. For each combination, DiaSynth was provided with the same 16 broad topics and tasked with generating 6 subtopics for each topic, followed by creating 6 personas for each subtopic. The statistics of the datasets generated using DiaSynth, including the number of dialogues, average number of turns, and average number of tokens per turn, are summarized in Table 3.2. All the experiments were run on a single A100 GPU with the generation time ranging from 2 hours to 4 hours.

3.3.1 Quality of the Dialogues

The quality of the synthetic datasets produced by DiaSynth was evaluated using FED, GPTScore, and G-Eval metrics, as detailed in Tables 3.3, 3.4, and 3.5. The results reveal distinct variations in performance across different model and dataset configurations, reflecting the unique characteristics of each.

3.3.1.1 Metric Scores

FED: The FED scores in Table 3.3 show that LLaMA-3 and GPT-4o achieve almost a perfect score (+1) in most of the criteria, while Phi-3 and InternLM-2.5 also have decent performances. GPT-4o has a clear advantage when it comes to generating likeable dialogues while there is not much separation on other criteria.

Criteria	Dialoguesum				Samsum			
	Phi-3	InternLM-2.5	LLaMA-3	GPT-4o	Phi-3	InternLM-2.5	LLaMA-3	GPT-4o
coherent	0.9536	0.8439	0.9684	0.9525	0.9161	0.8746	0.9829	0.9939
error recovery	0.9440	0.8313	0.9522	0.9407	0.9088	0.8647	0.9677	0.9876
consistent	0.9540	0.8359	0.9570	0.9417	0.9199	0.8734	0.9757	0.9878
diverse	0.9534	0.8353	0.9596	0.9423	0.9161	0.8655	0.9712	0.9836
depth	0.9521	0.8352	0.9592	0.9425	0.9130	0.8661	0.9731	0.9824
likeable	-0.0005	0.0048	0.0032	0.0121	-0.0004	0.0033	0.0003	0.0083
understand	0.9353	0.8278	0.9453	0.9368	0.9014	0.8582	0.9593	0.9788
flexible	3.96E-05	-0.0046	-0.0063	-0.0027	-0.0024	-0.0019	-0.0048	0.0004
informative	0.0009	0.0042	0.0063	0.0085	0.0034	0.0106	0.0100	0.0141
inquisitive	-0.0033	0.0069	0.0105	0.0144	-0.0040	0.0028	0.0029	0.006

TABLE 3.3: FED scores

GPTScore: Results illustrated in 3.4 are surprising in that GPT-4o is the worst performing model on GPTScore, which might require further research while LLaMA-3 clearly dominates the other models.

Criteria	Dialoguesum				Samsum			
	Phi-3	InternLM-2.5	LLaMA-3	GPT-4o	Phi-3	InternLM-2.5	LLaMA-3	GPT-4o
coherence	0.0286	0.0069	0.0189	0.0039	0.0325	0.0128	0.0288	0.0055
diversity	0.0310	0.0196	0.0430	0.0156	0.0372	0.0408	0.0655	0.0162
flexibility	0.0218	0.0084	0.0186	0.0059	0.0260	0.0194	0.0306	0.0094
understandability	0.0193	0.0061	0.0220	0.0039	0.0270	0.0174	0.0365	0.0084
inquisitiveness	0.0363	0.0244	0.0415	0.0180	0.0395	0.0504	0.0622	0.0186
consistency	0.0369	0.0137	0.0321	0.0080	0.0415	0.0306	0.0612	0.0119
informativeness	0.0172	0.0148	0.0318	0.0097	0.0201	0.0328	0.0542	0.0129
likeability	0.0213	0.0050	0.0110	0.0026	0.0232	0.0142	0.0168	0.0038
depth	0.0117	0.0080	0.0201	0.0053	0.0126	0.0177	0.0270	0.0079
error recovery	0.0342	0.0197	0.0440	0.0135	0.0290	0.0407	0.0558	0.0199

TABLE 3.4: GPTScore

G-Eval: Table 3.5 highlights GPT-4o’s dominance in engagingness and naturalness with perfect scores (3.0) for DialogSum, while InternLM-2.5 stands out in coherence (2.9990) and groundedness (2.9973) for DialogSum, and coherence (2.9983) and groundedness (2.9952) for SAMSum, suggesting it maintains high factual accuracy.

Dataset-Specific Performance. The contrasting performance of GPT-4o on the DialogSum and SAMSum datasets in Table 3.3 can be attributed to the differing structures of the dialogues in these datasets. DialogSum consists of more formal and structured dialogues, which aligns with the typical response style of GPT-4o, leading to its stronger performance. In contrast, SAMSum contains more casual, human-like conversations, which might explain GPT-4o’s relatively poorer performance, as it may not adapt as well to the informal, spontaneous nature of such dialogues. Overall, while GPT-4o excels in natural and engaging dialogue, LLaMA-3 offers the most versatility, and InternLM-2.5 provides a strong alternative with high coherence and groundedness.

	engagingness	naturalness	coherence	groundedness
DIALOGUESUM				
Phi-3	2.5236	2.7238	2.6308	2.5557
InternLM-2.5	2.9995	2.9989	2.9990	2.9973
LLaMA-3	2.9987	2.9988	2.9972	2.9935
GPT-4o	3	3	3	2.9975
SAMSUM				
Phi-3	2.4623	2.6821	2.5848	2.5060
InternLM-2.5	2.9992	2.9969	2.9983	2.9952
LLaMA-3	2.9976	2.9971	2.9969	2.9916
GPT-4o	2.9994	2.9977	2.9982	2.9944

TABLE 3.5: G-EVAL

3.3.1.2 Strong performance of LLaMA-3

The observed superiority of LLaMA-3 over GPT-4o is surprising because an 8 billion 8-bit quantized model not only competes with but also performs better than GPT-4o in certain metrics. We hypothesize that this could be due to the way GPT-4o was trained, which might make it more constrained in its responses, whereas LLaMA-3, being an open-source model, operates with fewer restrictions. This allows LLaMA-3 to exhibit greater flexibility, diversity, and adaptability in generating dialogues, potentially explaining its better performance in certain metrics. These characteristics can be seen in criteria like 'inquisitiveness' and 'likeability' in Table 3.4 and, 'depth' and 'diverse' in Table 3.3. These results suggest that for building human-like data generation frameworks, open-source LLMs are a more suitable choice than closed-source LLMs. The minimal constraints on response formatting during the training of open-source models enable them to generate more diverse, flexible, and human-like dialogues, making them better suited for tasks requiring natural and conversational interactions.

3.3.2 Fine-tuning and Performance Results

To validate the usability of the synthetic data generated using DiaSynth, we fine-tuned and evaluated several pretrained language models on the task of dialogue summarization. The summaries for dialogues generated by different LLMs were created using the corresponding LLMs through prompting. The pretrained models used for evaluation include DistilBART, BART, T5, and LED.

The experimental setup is designed as follows:

- Metrics are reported on the validation and test sets of DialogSum and SAM-Sum.
- To evaluate DiaSynth-generated data, we compared models fine-tuned on DiaSynth data with their base versions (no fine-tuning).
- In-domain training sets were randomly sampled to match the size of the DiaSynth-generated data, enabling fair comparison.
- The experiment aimed to quantify performance improvement of DiaSynth-fine-tuned models and assess alignment with in-domain data distributions.
- Models were fine-tuned for 2 epochs with a learning rate of $5e-5$ and a warmup of 50 steps.

Model	Before Finetuning			Finetuning on In-Domain Data		
	QAGS	BERTScore	ROUGE-L	QAGS	BERTScore	ROUGE-L
DIALOGSUM						
distillbart-cnn	0.6134	0.5093	0.1950	0.5586	0.7005	0.3367
bart-base	0.7007	0.5274	0.1375	0.4789	0.6868	0.2969
t5-base	0.5901	0.5491	0.1812	0.4766	0.6953	0.2986
led-base-16384	0.8261	0.5471	0.1634	0.4872	0.7084	0.3165
SAMSUM						
distillbart-cnn	0.6627	0.5500	0.2394	0.6041	0.6849	0.3578
bart-base	0.7563	0.4389	0.1765	0.5302	0.6520	0.3049
t5-base	0.5574	0.4190	0.1237	0.5460	0.6448	0.3000
led-base-16384	0.7429	0.4310	0.1812	0.5440	0.6522	0.3175

TABLE 3.6: Performance of models before and after finetuning on in-domain data

The results presented in Tables 3.6 and 3.7 present the performance of the base models, models finetuned on in-domain data and models finetuned on DiaSynth generated data. Models finetuned on DiaSynth data generally improves the performances from the BERTScore and ROUGE-L metrics. Surprisingly, for some models (LED and BART) the QAGS scores were higher than the models finetuned on DiaSynth. On further exploration, we found out that these models extracted multiple sentences from the given dialogue instead of generating a summary which led to high QAGS scores. Comparing models finetuned on in-domain data to those finetuned on DiaSynth data reveals that DiaSynth finetuning generally enhances factual accuracy, with BERTScore and ROUGE-L scores remaining comparable.

The disparity in BERTScore and ROUGE-L results may be due to format variations. Models fine-tuned on in-domain data were evaluated on summaries that matched the training format closely, while DiaSynth-fine-tuned models were trained on LLM-generated summaries and evaluated on human-generated summaries, leading to minor format mismatches. Comparison between the different LLMs from Table 3.7, shows that GPT-4o is better at generating dialogues and summaries that are formal in nature while LLaMA-3 and open source LLMs would be better for generating dialogues that are informal and casual in nature.

Model	Phi-3			InternLM-2.5			LLaMA-3			GPT-4o		
	QAGS	BERTScore	ROUGE-L	QAGS	BERTScore	ROUGE-L	QAGS	BERTScore	ROUGE-L	QAGS	BERTScore	ROUGE-L
DIALOGUESUM												
distillbart-cnn	0.6588	0.5778	0.2187	0.6420	0.6008	0.2167	0.6586	0.6161	0.2040	0.6713	0.6242	0.2014
bart-base	0.5355	0.5958	0.2029	0.5418	0.6212	0.1897	0.5825	0.6033	0.1789	0.5590	0.6039	0.1769
t5-base	0.5937	0.5949	0.2047	0.5825	0.5941	0.1878	0.6034	0.6172	0.1959	0.6305	0.6319	0.2044
led-base-16384	0.5358	0.6129	0.2109	0.5189	0.6027	0.1606	0.5697	0.6302	0.1999	0.5791	0.6308	0.1989
SAMSUM												
distillbart-cnn	0.6585	0.5931	0.2262	0.6388	0.6066	0.2422	0.6849	0.6029	0.2374	0.6757	0.6029	0.2291
bart-base	0.5648	0.5665	0.2146	0.5435	0.5663	0.2021	0.6132	0.5899	0.2345	0.5707	0.5808	0.2154
t5-base	0.5905	0.5397	0.2085	0.5457	0.5193	0.1854	0.6412	0.5054	0.1976	0.6023	0.5419	0.1979
led-base-16384	0.5883	0.5477	0.2289	0.5457	0.5615	0.2167	0.5917	0.5785	0.2390	0.5738	0.569	0.2298

TABLE 3.7: Performance after finetuning on synthetic data

To assess the percentage improvement and percentage coverage of the distributional characteristics of the in-domain data by the synthetically generated data, we use Equations 3.1a and 3.1b respectively. We use the scores of models finetuned on LLaMA-3 generated data because of its dominance in both quality and usability. Across the 24 reported results, the overall coverage percentage of the LLaMA-3 generated data is **90.48%**. Notably, the QAGS scores of models fine-tuned on synthetic data surpass those of models trained on in-domain data, suggesting that synthetic data can match or even exceed in-domain data performance in some aspects. Excluding QAGS, the coverage percentage is calculated to be **77.07%**. In addition to the average percentages, we also present the model wise percentage improvement and coverage in Table 3.8 and 3.8. The results presented are with respect to the dialogues generated using LLaMA-3 and they illustrate clear improvements for every model, highlighting that even with moderate LLMs of small scale (3B - 8B), high-quality synthetic dialogue datasets can be effectively created across different domains and different dialogue formats.

$$\% \text{ Improvement} = \frac{\text{After Finetuning on synthetic data} - \text{Before finetuning}}{\text{Before finetuning}} \quad (3.1a)$$

$$\% \text{ Coverage} = \frac{\text{Score After Finetuning on DiaSynth data}}{\text{Score After Finetuning on In-domain data}} \quad (3.1b)$$

Model	Dialoguesum		SAMSum	
	% Improvement	% Covered	% Improvement	% Covered
distilbart-cnn	10.96	88.81	6.07	87.25
bart-base	9.21	90.6	16.12	94.35
t5-base	7.59	93.67	30.04	87.36
led-base-16384	2.14	89.68	15.25	90.91

TABLE 3.8: Summarization results for Dialoguesum and SAMSum

3.3.3 Additional Downstream Task: Response Generation

Model	Before Fine-tuning	In-domain Data	Llama3	GPT-4o
t5-base	0.4003	0.6870	0.6572	0.6612
bart-base	0.5681	0.6875	0.6721	0.6630

TABLE 3.9: BERTScore evaluation of response generation models fine-tuned on synthetic and in-domain datasets.

To further evaluate the utility of DiaSynth-generated data, we conducted an additional downstream task: response generation. This task was included to validate the effectiveness of our synthetic data beyond summarization. For these experiments, we selected Llama3 and GPT-4o as base datasets since they demonstrated superior performance across both quality and summarization metrics. We employed BERTScore as our evaluation metric due to its effectiveness in measuring the similarity between generated and reference responses. The results are presented in Table 3.9.

Both models show a significant improvement in BERTScore after fine-tuning on DiaSynth-generated data compared to the pre-trained baseline, highlighting its effectiveness in enhancing model performance. Additionally, models fine-tuned

on DiaSynth data achieve scores that are **close to those fine-tuned on in-domain data**. For instance, **t5-base** achieves a BERTScore of **0.6572** on Llama3-generated data and **0.6612** on GPT-4o-generated data, compared to **0.6870** for in-domain fine-tuning. These results indicate that DiaSynth-generated data serves as a viable alternative for fine-tuning response generation models, performing comparably to in-domain data, even in low-resource scenarios.

3.4 Ablations and Hallucination Study

3.4.1 Ablation study

To further validate the effectiveness of the DiaSynth framework, we conducted ablation studies by evaluating the impact of removing sub-topics and personas from the data generation pipeline. The goal was to assess their contribution to the quality of the generated dialogues.

For these experiments, we generated approximately 960-1000 dialogues using Phi-3 and compared three settings:

- **subtopics**: Removing sub-topics while keeping personas.
- **personas**: Removing personas while keeping sub-topics.
- **diasynth**: The full DiaSynth-generated data with both personas and sub-topics.

The ablation studies presented in Tables 3.10 and 3.11 demonstrate that the inclusion of both personas and sub-topics significantly enhances the quality of generated dialogues across FED and GPTScore metrics. For both DialogSum and SAMSum few-shot examples, dialogues generated with the full DiaSynth framework, incorporating both personas and sub-topics, achieved the highest scores in coherence, diversity, and consistency. This indicates that structured dialogue generation benefits from incorporating diverse sub-topics while maintaining persona-driven consistency.

An interesting observation arises in Table 3.10, where for the SAMSum dataset, the best-performing configuration involved using only sub-topics without personas.

Metric	Dialoguesum			SAMSum		
	subtopics	personas	diasynth	subtopics	personas	diasynth
Coherent	0.9252	0.9584	0.9536	0.0106	0.0109	0.0325
Error Recovery	0.9022	0.9414	0.944	0.0323	0.0213	0.029
Consistent	0.9095	0.9512	0.954	0.0192	0.0415	0.037
Diverse	0.9139	0.9512	0.9534	0.0224	0.0191	0.0372
Depth	0.9193	0.9533	0.9521	0.0187	0.0158	0.0126
Likeable	0.0069	0.007	-0.0005	0.0128	0.0075	0.0232
Understandable	0.8918	0.9339	0.9353	0.0159	0.0252	0.0415
Flexible	-0.0038	-0.0042	0	0.0235	0.0141	0.0395
Informative	0.0096	0.0072	0.009	0.0228	0.0155	0.021
Inquisitive	0.0129	0.0063	-0.0033	0.0334	0.0137	0.0415

TABLE 3.10: FED scores with DialogSum and SAMSum as base

Metric	Dialoguesum			SAMSum		
	subtopics	personas	diasynth	subtopics	personas	diasynth
Coherence	0.0118	0.0098	0.0286	0.0106	0.0109	0.0325
Diversity	0.0246	0.0198	0.031	0.0224	0.0191	0.0372
Flexibility	0.0239	0.0121	0.0193	0.0235	0.0141	0.0395
Understandable	0.0137	0.0121	0.0363	0.0159	0.0252	0.0415
Inquisitive	0.0358	0.0308	0.0363	0.0334	0.0137	0.0415
Consistent	0.0175	0.0132	0.0369	0.0192	0.0415	0.037
Informative	0.0242	0.0155	0.0201	0.0228	0.0155	0.021
Likeability	0.0083	0.0076	0.0213	0.0128	0.0075	0.0232
Depth	0.0216	0.0181	0.0117	0.0187	0.0158	0.0126
Error Recovery	0.0337	0.0249	0.0342	0.0323	0.0213	0.029

TABLE 3.11: GPTScore with DialogSum and SAMSum as base

This deviation can likely be attributed to the more informal nature of SAMSum dialogues, where structured personas introduce a formal communication style that does not align well with the dataset. In contrast, in more structured datasets like DialogSum, the addition of personas provides clear improvements, ensuring dialogue coherence and natural flow.

Moreover, the largest improvements in quality are seen in coherence, error recovery, and understandability, particularly when both sub-topics and personas are included. While sub-topics alone contribute significantly to improving diversity and depth, their combination with personas enhances overall dialogue quality. This suggests that dataset characteristics play a crucial role in determining the effectiveness of persona modeling, highlighting the need for adaptive strategies in synthetic dialogue generation. Ultimately, these findings reinforce that DiaSynth-generated dialogues are robust and adaptable, providing high-quality synthetic data across both structured and informal conversational settings.

3.4.2 Hallucination study

In addition to evaluating the quality and usability of dialogues produced by DiaSynth, we conducted a study on the phenomenon of hallucinations within the generated dialogues. Hallucinations in language models refer to instances where the output contains misleading or incorrect information or situations where the model repeats the same content. To evaluate the occurrence of hallucinations, we compared the generated dialogues with their respective summaries and assessed them using two well-known hallucination benchmarks: **SelfCheckGPT** [40] and **ChainPoll** [41]. This analysis provides insights into the prevalence of hallucinations and informs strategies for improving dialogue quality in future iterations of DiaSynth. The results are presented in Table 3.12.

3.4.2.1 SelfCheckGPT

SelfCheckGPT quantifies the self-consistency of LLM outputs by examining agreement across multiple outputs from the same prompt. This assessment reveals potential inaccuracies through metrics like SelfCheck-BertScore.

The SelfCheck-BERTScore results for various models show that hallucination levels are at worst around 25%, which is acceptable but still indicates areas for improvement. Across both datasets, **Phi-3** demonstrates the most robustness, likely due to its pretraining on structured, textbook-like data, which may contribute to greater consistency and factual accuracy.

3.4.2.2 ChainPoll

ChainPoll utilizes a chain-of-thought prompting approach to identify hallucinations by iteratively polling the model with structured reasoning prompts. This method systematically detects both open-domain and closed-domain hallucinations, where lower scores indicate fewer hallucinations.

The ChainPoll scores indicate that hallucination levels on these models are generally low, with the best performance seen by **GPT-4o** on SAMSum, which achieves the lowest score of 0.120, suggesting minimal hallucinations. On the other hand,

LLaMA-3 scores higher at 0.237 on SAMSum, indicating more frequent hallucinations. These findings highlight different models’ strengths in generating accurate and reliable dialogues.

3.4.2.3 Implications for DiaSynth

The results from both SelfCheckGPT and ChainPoll evaluations suggest that DiaSynth, when leveraging models like Llama 3, is capable of generating dialogues with relatively low hallucination rates. However, specific models show variability in performance across datasets, indicating that further enhancements, such as fine-tuning or incorporating additional guardrails, could improve DiaSynth’s robustness in generating reliable dialogues across diverse domains.

LLM	DialogSum		SAMSum	
	ChainPoll	SCGPT-BERTScore	ChainPoll	SCGPT-BERTScore
Phi-3	0.198	0.791	0.154	0.785
InternLM-2.5	0.199	0.726	0.159	0.716
LLaMA-3	0.205	0.793	0.237	0.733
GPT-4o	0.178	0.765	0.120	0.742

TABLE 3.12: Hallucination calculation for DialogSum and SAMSum few-shot data

3.5 Summary

In this chapter, we introduced *DiaSynth*, a framework for generating synthetic dialogues to address the lack of domain-specific training data for dialogue systems. DiaSynth uses a step-by-step process that includes generating subtopics, creating personas, and using CoT reasoning to guide dialogue generation. This process helps create realistic and varied dialogues that reflect different settings, tones, and conversation styles.

Our experiments showed that models fine-tuned on DiaSynth-generated data performed significantly better on dialogue summarization tasks compared to base models. LLaMA-3 worked well for informal conversations, while GPT-4o was better for structured ones. These findings highlight DiaSynth’s usefulness for generating training data in low-resource domains, making it a valuable tool for building more effective dialogue systems.

Chapter 4

CS-Sum: A Benchmark for Code-Switching Dialogue Summarization and the Limits of Large Language Models

In this chapter, we introduce CS-Sum - a code-switch dialogue summarization benchmark which consists of 1.4K dialogues in three language pairs - English - Mandarin (EN-ZH), English - Tamil (EN-TA) and English - Malay (EN-MS). CS-Sum benchmark is introduced to evaluate the CS comprehension of the current SOTA LLMs (less than 10B parameters). This chapter shows that though the LLMs perform well on monolingual data, they make subtle errors when handling CS dialogues that completely alter the meaning of the dialogues. This chapter is organised as follows: Section 4.1 details our data curation process, Section 4.2 presents the performances of the LLMs in various settings on CS-Sum, Section 4.3 gives a detailed analysis and examples of the different types of errors the LLMs make when handling CS dialogues and finally Section 4.4 provides the summary of the chapter.

4.1 CS-Sum Benchmark

The goal of building the CS-Sum benchmark, was to address the lack of a benchmark that evaluates the CS comprehension of LLMs. We selected Mandarin (zh), Bahasa Melayu (ms), and Tamil (ta) because these languages have large bilingual speaker communities that frequently code-switch with English. To efficiently create a diverse CS benchmark within resource constraints, we translated a subset of the combined test sets from DialogSum and SAMSum. We recruited 7 native speakers for each language to translate the English dialogues to CS dialogues in their respective languages. The translators were native speakers, all university students (bachelor’s or master’s level), with each assigned 200 dialogues. The entire translation process lasted for about 5 months.

4.1.1 CS Metrics

In this section we provide a brief explanation on metrics that are used to quantify the amount of code switching in a given corpus and report those metrics on the CS-Sum benchmark. The five most common CS metrics [42] that are used to measure the amount of code-switching in a corpus are:

- **M-Index:** The M-Index measures the balance of language distribution in a corpus. Ranging from 0 (monolingual) to 1 (equal distribution), higher values indicate more balanced language representation.
- **I-Index (Integration Index):** The I-Index quantifies the probability of language switching between tokens. It ranges from 0 to 1, where 0 indicates no switching and 1 represents frequent alternations between languages.
- **Burstiness:** Burstiness measures the irregularity of code-switching. Ranging from -1 to 1, -1 indicates regular switching and 1 suggests bursty, clustered switching events.
- **Span Entropy:** Span Entropy measures the unpredictability of monolingual segments between code-switches. Ranging from 0 to $\log_2(M)$, higher values indicate more variation in segment lengths, making the switching pattern less predictable.

Language	M-Index	I-Index	Burstiness	Span Entropy	Memory
EN-ZH	0.40	0.36	-0.79	0.13	-0.08
EN-MS	0.41	0.38	-0.65	0.64	-0.24
EN-TA	0.42	0.40	-0.68	0.58	-0.22

TABLE 4.1: CS metrics measured on the CS-Sum dataset

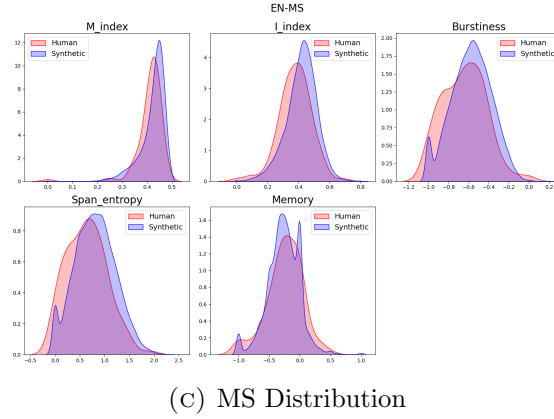
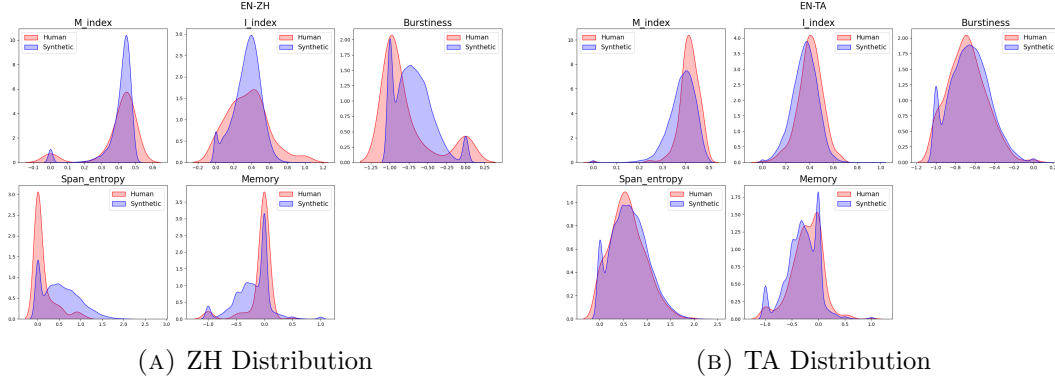
- **Memory:** Memory quantifies the autocorrelation of consecutive language spans. Ranging from -1 to 1, -1 indicates alternating span lengths, while 1 shows similar lengths between consecutive spans, reflecting the temporal structure of switching.

We present the CS metrics [42] on CS-Sum in Table 4.1. The values show that CS-Sum exhibits a moderate level of code-switching, with the M-Index ranging from 0.40 to 0.42 across the three languages, indicating that switching occurs frequently but remains balanced within the dialogues. The I-Index values (0.36 to 0.40) further suggest that code-switching often happens within sentences, with EN-TA showing a slightly higher degree of intra-sentential switching compared to EN-ZH and EN-MS. The burstiness values, which range from -0.79 for EN-ZH to -0.65 for EM-MS, demonstrate that code-switching tends to occur in bursts, with longer monolingual segments followed by intense switching. The span entropy values indicate that code-switching is more predictable in EN-ZH (0.13) but less so in EN-MS (0.64) and EN-TA (0.58), suggesting that LLMs will face greater difficulty in anticipating switch points in the latter two languages. Lastly, the memory values (-0.08 for EN-ZH, -0.24 for EN-MS, and -0.22 for EN-TA) show that monolingual stretches of language persist before switching, which implies that models must carefully track these segments and manage transitions between languages in the summarization process.

4.1.2 Gemini generated training data

To check if training on synthetic CS data improves the LLMs’ performance on CS-Sum, we merged the training sets of DialogSum and SAMSum which resulted in 19014 pairs of English dialogue and summary. The merged set was converted to CS dialogues (EN-TA, EN-MS, EN-ZH) by using Gemini-2 ¹ [43] which we call

¹<https://ai.google.dev/>, gemini-2-flash-exp



CS-Sum-Syn. To compare the difference in distribution between CS-Sum and the CS-Sum-Syn, we present the CS metrics for the three languages in Figures 4.1a, 4.1b and 4.1c. From the figures, it can be seen that the CS metrics distribution of CS-Sum-Syn for EN-MS and EN-TA languages are similar to CS-Sum (human generated) while there is a significant difference for EN-ZH likely because Gemini-2 produces more unpredictable switching (higher span entropy), less bursty transitions, and longer monolingual spans, suggesting that it struggles with natural EN-ZH alternation and tends to insert CS at unnatural positions.

The conversion process involved using the following prompt to generate CS dialogues for three languages, augmented with few-shot examples. The prompt ensures that the code-switched dialogues follow the structure of the original English dialogues:

CS pair	KL Div.	JS Div.
EN-ZH	2.4786	0.4918
EN-MS	0.5481	0.1454
EN-TA	0.5089	0.1017

TABLE 4.2: Divergence

Example Prompt for Malay-English Code-Switching

You are a {Malay/Tamil/Chinese} person in your 20s.
 You are recruited for translating English dialogues to English-{Malay/Tamil/Chinese} code-switched dialogues.
 The code-switched dialogues should follow the same structure as the English dialogue.
 This is for educational purposes, so DO NOT include swear words in your translation.
 Return just the code-switched dialogue.

To quantify the difference in the distributions we measured the Jensen-Shannon divergence [44] and KL divergence [45] between CS-Sum and CS-Sum-Syn for each CS language pair which is presented in Table 4.2. The higher values for EN-ZH compared to the other language pairs, reinforce the inference from Figure 4.1a. The synthetic data generation process is described in Appendix ??.

4.2 LLM performance on CS-Sum

In this section we present the performance of ten LLMs (9 open-source² and 1 proprietary LLMs) on CS-Sum under different settings, namely, Few-Shot, Translate-Summarize, LoRA and QLoRA. In LoRA and QLoRA settings, the open-source models are finetuned on CS-Sum-Syn. The open-source LLMs that are presented are LLaMA-3-8B [46], Mistral-7B [47], Ministral-8B³, Gemma-2-2B and Gemma-2-9B [48], Qwen2.5-2B and Qwen2.5-7B [49], SEA-Lion-Gemma-9B⁴ and SEALLM-7B [50]. The proprietary LLM that is used is GPT-4o [51].

²Intruction tuned versions

³huggingface.co/mistralai/Ministral-8B-Instruct-2410

⁴<https://sea-lion.ai/>

These LLMs were selected due to their multilingual exposure during pretraining and their diverse parameter sizes, ranging from 2B to 9B. Open-source models such as LLaMA-3-8B, Mistral-7B, and Gemma-9B have demonstrated strong generalization across NLP tasks, while Qwen2.5-7B, SEA-Lion-Gemma-9B and SEALLM-7B were included for their explicit focus on multilinguality, particularly in handling languages like Bahasa Melayu, Tamil and Mandarin. A proprietary model like GPT-4o serve as strong baseline given its SOTA performance in cross-lingual comprehension.

By evaluating models across different scales and varying levels of multilingual exposure during pretraining, we aim to analyze the impact of model size, multilingual training, and task-specific fine-tuning on their ability to handle CS. LoRA and QLoRA fine-tuning on CS-Sum-Syn improves model performance on automated metrics, indicating better alignment with CS summaries. However, it remains unclear whether this improvement reflects genuine CS understanding or simply a learned ability to replicate synthetic CS patterns. To investigate this, we present a detailed error analysis in Section 4.3 which shows that finetuning on synthetic data does not improve the CS comprehension of the LLMs and in some cases even worsens the comprehension.

4.2.1 Evaluation Metrics

The metrics that have been used for evaluating the generated summaries are Rouge-L [35], BERTScore [34], SBERT-Cosine [52], Jaccard and Meteor [53]. The following subsections describe SBERT-Cosine, Jaccard and Meteor. Section 3.2.2.2 provides a detailed description on BERTScore.

4.2.1.1 SBERT-Cosine

SBERT (Sentence-BERT) [52] is an extension of BERT that uses a siamese and triplet network structure to derive semantically meaningful sentence embeddings. The embeddings are fine-tuned to capture sentence-level semantics, allowing them to be compared using similarity measures like cosine similarity. This method overcomes the computational inefficiency of traditional BERT models, where comparing large numbers of sentence pairs would require excessive computation.

SBERT uses cosine similarity to evaluate the semantic similarity between sentence pairs. The embeddings are generated by feeding sentences into a siamese network, where both sentences share the same weight parameters, allowing for efficient and high-quality similarity measurements. It has been shown to significantly improve performance in semantic textual similarity (STS) tasks compared to previous methods like GloVe embeddings and Universal Sentence Encoder.

The cosine similarity between two vectors u and v is computed as:

$$\text{cosine_similarity}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

where u and v are the embeddings of the two sentences, and $\|\cdot\|$ denotes the Euclidean norm of the vector.

SBERT has been evaluated on various tasks, including Semantic Textual Similarity (STS) and SentEval, where it outperforms other state-of-the-art methods in many cases. It also shows notable improvements when used in large-scale semantic search tasks and clustering, making it a robust tool for applications requiring efficient sentence similarity computation.

4.2.1.2 Jaccard

The **Jaccard Index** is a statistical measure used to evaluate the similarity and diversity of sample sets. It is defined as the size of the intersection divided by the size of the union of the two sets. The Jaccard Index ranges from 0 to 1, where 0 indicates no similarity and 1 indicates that the two sets are identical.

For two sets A and B , the Jaccard index is calculated as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Where: - $|A \cap B|$ is the size of the intersection of sets A and B . - $|A \cup B|$ is the size of the union of sets A and B .

In the context of text similarity, the Jaccard index can be used to compare the similarity between two pieces of text by treating them as sets of words or tokens.

Specifically, the two texts are tokenized, and the Jaccard index is computed based on the set of words (or tokens) present in each text.

The Jaccard metric is often used in tasks like document similarity, clustering, and classification, where the goal is to measure how similar two documents are based on their content. It is particularly useful when the order of the words does not matter, as it focuses solely on the presence or absence of shared elements.

4.2.1.3 Meteor

The METEOR metric [53] is an automatic evaluation measure designed to improve upon the weaknesses of BLEU, particularly by addressing issues with recall and word order. METEOR is based on a generalized concept of unigram matching between a machine-generated translation and reference translations. The metric goes beyond simple word matching by incorporating stemming, synonym matching, and word order considerations, allowing for a more flexible and robust evaluation.

The METEOR score is calculated through a multi-stage process, where unigrams are first matched between the two sentences based on exact matches, stemmed forms, and synonyms. The metric uses the harmonic mean of unigram precision and recall, which places greater weight on recall to account for the completeness of the translation. To further improve the evaluation, a penalty is applied for word order fragmentation, which captures how well-ordered the matched words are in the machine-generated translation compared to the reference.

The METEOR score is computed using the following formula:

$$\text{METEOR} = F_{\text{mean}} \times (1 - \text{Penalty})$$

Where: F_{mean} is the harmonic mean of unigram precision (P) and recall (R). The Penalty term penalizes translations with higher fragmentation, which is calculated based on the number of chunks of matched words. The fewer the chunks, the lower the penalty.

4.2.2 Few-shot and Translate-Summarize

In this subsection, we describe the experimental setup for evaluating LLMs on CS-Sum under the *Few-Shot* and *Translate-Summarize* paradigms. These experiments assess how well LLMs generate accurate and natural code-switched (CS) summaries without explicit fine-tuning.

Few-shot In the Few-Shot setup, we provide each LLM with three human-crafted examples containing code-switched dialogues and their corresponding summaries. The results from this setup, will quantify the inherent CS understanding of the LLMs for the three language pairs.

Translate-Summarize In the Translate-Summarize approach, each CS dialogue is first translated into English before being summarized. This serves as a baseline to determine whether the presence of code-switching affects summary quality. The translation step is performed by the same LLM, ensuring that the summarization model receives only monolingual input.

4.2.3 LoRA and QLoRA Fine-Tuning for CS Dialogue Summarization

In this subsection, we describe the fine-tuning process using *LoRA* (Low-Rank Adaptation) [54] and *QLoRA* (Quantized Low-Rank Adaptation) [55] to adapt open-source LLMs for CS dialogue summarization. Full fine-tuning of LLMs is computationally expensive, so we utilize PEFT techniques, which reduce the number of parameters that need to be updated, while maintaining or even improving performance. Specifically, we explore whether task-specific adaptation improves the ability of the models to understand and generate CS dialogue.

For this experiment, the models were fine-tuned on the CS-Sum-Syn dataset. The goal was to determine if LoRA and QLoRA could effectively adapt these LLMs for CS comprehension without requiring extensive computational resources. The fine-tuning process follows specific hyperparameter configurations, which are summarized in Table 4.3.

The key hyperparameters for the fine-tuning process include:

- **Rank (r)**: This parameter controls the low-rank decomposition of the model’s weight matrices.
- **Scaling Factor (α)**: This factor controls the scale of the low-rank adaptation.
- **Learning Rate (lr)**: The learning rate for training the adapted model.

All models were trained for 4 epochs with a batch size of 8, using gradient accumulation for 8 steps. The training process utilized a cosine learning rate schedule after a 3% warm-up. We applied bf16 precision and gradient checkpointing for memory-efficient training. The models were trained on an H100 GPU, with training times varying between 1 to 4 hours, depending on the architecture and size of the LLM.

The fine-tuning hyperparameters for each model are summarized in Table 4.3. These settings allow for efficient and effective fine-tuning across various LLMs, enabling their adaptation for CS dialogue summarization tasks.

Model	r	α	LR
Ministral-8B	32	32	5e-5
LLaMA-3-8B	16	16	5e-5
Qwen2.5-3B	64	64	3e-5
Gemma-2-9B	32	32	5e-5
SEA-Lion-Gemma-9B	32	32	5e-5
Gemma-2-2B	8	8	5e-5
Qwen-2.5-7B	64	64	3e-5
Mistral-7B	32	32	5e-5
SEALLM-7B	64	64	3e-5

TABLE 4.3: Hyperparameters used for LoRA and QLoRA fine-tuning.

4.2.4 Result Analysis

In this subsection, we present the results of the LLMs’ performance on the CS-Sum using standard summarization metrics like ROUGE [35], BERTScore [34], SBERT-Cosine [52], Jaccard and METEOR [53]. This comprehensive set of metrics evaluate both word-level matching and semantic similarity.

Model	Lang	ROUGE-L	BERTScore	SBERT-COSINE	JACCARD	METEOR
Gemma-2-2B	EN-ZH	0.2330 / -4	0.8876 / -0.35	0.7268 / -1.40	0.1658 / -7.64	0.2699 / -0.9
	EN-TA	0.2202 / -7	0.8812 / -0.38	0.6755 / -0.85	0.1523 / -9.19	0.2453 / -5.23
	EN-MS	0.2421 / -7.18	0.8898 / -0.98	0.7352 / -4.35	0.1739 / -13.52	0.2902 / -12.43
Qwen2.5-3B	EN-ZH	0.2393 / 9.88	0.8824 / 0.93	0.7143 / 1.70	0.1643 / 5.82	0.2974 / 2.05
	EN-TA	0.2227 / -1.46	0.8805 / 0.44	0.6880 / -2.35	0.1468 / -0.13	0.2591 / -4.82
	EN-MS	0.2408 / 6.21	0.8868 / 0.60	0.7308 / -2.25	0.1694 / 1.14	0.3100 / -4.57
Qwen2.5-7B	EN-ZH	0.2482 / 27.10	0.8106 / 11.20	0.4927 / 54.12	0.0783 / 162.62	0.1118 / 188.89
	EN-TA	0.2800 / -5.11	0.8937 / -0.55	0.7081 / -2.14	0.1837 / -9.62	0.2723 / -9.20
	EN-MS	0.3117 / -1.65	0.9041 / -0.46	0.7520 / -0.67	0.2122 / -8.37	0.3372 / -10.26
SEALLM-7B	EN-ZH	0.2115 / 19.25	0.8122 / 9.15	0.4836 / 45.48	0.0834 / 89.27	0.1297 / 104.29
	EN-TA	0.2270 / 0.99	0.8686 / 1.21	0.6470 / 2.53	0.1336 / 3.44	0.2287 / 1.55
	EN-MS	0.2575 / 8.41	0.8801 / 1.28	0.6956 / 4.91	0.1633 / 9.04	0.2800 / 5.95
Mistral-7B	EN-ZH	0.2568 / -3.10	0.8902 / -0.42	0.7208 / -0.34	0.1754 / -8.01	0.3010 / -7.82
	EN-TA	0.2198 / -2.15	0.8792 / 0.04	0.6847 / -0.95	0.1470 / -6.91	0.2657 / -10.45
	EN-MS	0.2421 / 1.59	0.8897 / -0.46	0.7293 / -2.15	0.1796 / -11.49	0.3084 / -8.40
Ministral-8B	EN-ZH	0.2560 / 4.23	0.8875 / 0.52	0.7037 / 1.06	0.1588 / 6.13	0.2517 / 1.15
	EN-TA	0.2468 / -5.61	0.8858 / -0.18	0.6752 / -2.95	0.1561 / -3.66	0.2387 / -8.53
	EN-MS	0.2700 / 6.97	0.8917 / 0.37	0.7262 / 0.03	0.1794 / 2.93	0.2843 / -0.09
LLaMA-3-8B	EN-ZH	0.2868 / -2.29	0.8845 / 0.76	0.7090 / 4.09	0.1918 / -6.99	0.2916 / 0.88
	EN-TA	0.2453 / 0.47	0.8769 / 0.99	0.6707 / 3.28	0.1598 / 1.07	0.2570 / -2.13
	EN-MS	0.2599 / 8.06	0.8724 / 2.04	0.6538 / 11.21	0.1804 / 0.02	0.2833 / 5.50
Gemma-2-9B	EN-ZH	0.2995 / -5.62	0.8987 / -0.27	0.7603 / -0.91	0.2054 / -5.83	0.3246 / -3.06
	EN-TA	0.2761 / -3.85	0.8911 / 0.57	0.7203 / 4.42	0.1832 / 4.63	0.3107 / 1.25
	EN-MS	0.3009 / -3.00	0.8975 / 0.01	0.7573 / -0.94	0.2096 / -3.96	0.3279 / -0.07
SEA-Lion-Gemma-9B	EN-ZH	0.2799 / -3.96	0.8968 / -0.34	0.7542 / -0.41	0.1820 / -4.29	0.2944 / -4.46
	EN-TA	0.2703 / 0.26	0.8898 / 0.48	0.7138 / 3.79	0.1727 / -0.12	0.2884 / -2.38
	EN-MS	0.2860 / -1.56	0.8963 / -0.12	0.7404 / 1.21	0.1837 / -4.16	0.2983 / -4.75
GPT-4o	EN-ZH	0.2965 / -1.03	0.8816 / 2.21	0.7082 / 9.63	0.1792 / 11.66	0.2853 / 16.52
	EN-TA	0.3157 / -4.92	0.8981 / 0.39	0.7570 / 2.19	0.2041 / -0.55	0.3419 / 0.75
	EN-MS	0.3126 / -1.70	0.9037 / -0.15	0.7750 / 0.12	0.2045 / 0.33	0.3365 / 1.09

TABLE 4.4: Few-Shot Performance/Translate-Summarize % improvement on CS-Sum

The Few-Shot performance of the LLMs across the three CS language pairs is shown in Table 4.4. The high scores of semantic similarity metrics like BERTScore and SBERT-Cosine is misleading, because the summaries generated by the LLMs do not convey the actual content of the conversation. Figure 4.2 shows a summary generated by Gemma-2-9B for a EN-TA dialogue from CS-Sum. The ROUGE-L scores of the LLMs are almost half of the best performing models on the monolingual DialogSum⁵ and SAMSum⁶ benchmarks.

From Table 4.4, we observe that GPT-4o is the best performing model, but we cannot give further reasoning on their performance due to the lack of transparency on how the model was developed. Gemma-2-9B consistently outperforms other open-source models across most metrics and language pairs. This aligns with findings from the Gemma-2 paper [48], which highlights that the model was trained with

⁵<https://paperswithcode.com/sota/text-summarization-on-dialogsum>⁶<https://paperswithcode.com/sota/text-summarization-on-samsum-corpus>

extensive multilingual data and an optimized tokenizer designed to improve language generalization. Despite not being specifically optimized for code-switching, Gemma-2’s exposure to a diverse multilingual corpus enables it to handle mixed-language text better than models with SEA-specific pretraining. In contrast, SEA-Lion-Gemma-9B lags slightly behind, despite its specialization in SEA languages.

Interestingly most of the models underperform on EN-ZH, which was surprising given that all the models were exposed to ZH data during their pretraining. By going through the summaries generated by different LLMs are because some of the summaries generated by the LLMs are in Chinese, while it was specifically prompted to generate the summary in English. This has led to their lower scores in the EN-ZH pair in automated metrics but in the our finegrained analysis (Section 4.3), LLMs usually have smaller error rates on EN-ZH in comparison to other CS pairs.

Additionally, the smaller models, such as Gemma-2-2B and Qwen2.5-3B, exhibit noticeably weaker performance across all language pairs compared to the bigger LLMs belonging to the same family, Gemma-2-9B and Qwen2.5-7B respectively, showing that performance increases with scale.

Another interesting result from Table 4.4, is that the performance of SEALLM-7B is comparable to smaller and non-SEA specific models such as Gemma-2-2B and Qwen2.5-3B. Comparing the summaries generated we found that the in-context learning ability of SEALLM is worse than that of the smaller models. In the few-shot setting SEALLM-7B exhibits the following problems: **(a)** extracts utterances from the dialogue, **(b)** describes the dialogue instead of summarizing and **(c)** tries to generate the summary in languages other than English. These issues have led to its lower performance on the CS-Sum benchmark.

Translate-Summarize % Improvement Table 4.4 presents the percentage improvement in performance under the Translate-Summarize setting. Ideally, if LLMs were capable of accurately translating CS dialogues into English without losing key contextual information, we would expect a significant performance boost in this setting. However, the results indicate otherwise—most models either exhibit no improvement or even a decline in performance across all metrics.

LLM	Human
Emily is asking where the ASEES conference is held. Chloe thinks it's probably in San Francisco or in Hawaii, but James thinks it's held in California. Chloe thinks that's sad.	James informs that the next ASEEES conference will be held in San Francisco. Chloe was hoping for Hawaii. James doesn't think there will be a conference in Hawaii soon due to the high cost of travel.

FIGURE 4.2: Wrong summary with high BERTScore 0.903

This suggests that rather than improving the summarization process, the translation step introduces additional challenges, likely due to the models' inability to correctly interpret and preserve the linguistic structures present in CS dialogues. Notably, models like Gemma-2-2B and Qwen2.5-3B show a drop of up to 7% in ROUGE-L, with even larger declines in Jaccard and METEOR, indicating a loss of essential meaning when translating CS inputs.

Interestingly, Qwen2.5-7B and SEALLM-7B show gains in EN-ZH (27.1% and 19.25% in ROUGE-L, respectively), suggesting that models with explicit Mandarin pretraining (Qwen) or SEA exposure (SEALLM) may perform better at translating Mandarin-English CS dialogues. However, these improvements are not observed in Tamil-English or Malay-English, reinforcing the idea that CS translation remains highly unreliable outside of high-resource languages. The fact that most models exhibit no improvement or even a decline in performance further confirms that current LLMs struggle not only with summarizing CS dialogues but also with accurately translating them. This drop in performance suggests that the translation step introduces additional challenges, likely due to the models' inability to preserve key discourse elements and linguistic structures present in CS dialogues. The failure to retain essential meaning across translation highlights a fundamental gap in CS comprehension, as LLMs appear to lack the necessary representations to model the complex interleaving of languages within conversations.

LoRA and QLoRA Results Table 4.5 presents the percentage improvements from applying LoRA and QLoRA fine-tuning techniques using synthetic data (CS-Sum-Syn) over the Few-Shot baseline on the CS-Sum benchmark. The primary

Model	Lang	ROUGE-L	BERTScore	SBERT-COSINE	JACCARD	METEOR
Gemma-2-2B	EN-ZH	9.13 / 22.26	0.15 / 1.21	3.52 / 6.23	19.28 / 21.69	15.47 / 15.78
	EN-TA	21.52 / 23.29	1.62 / 1.79	12.10 / 11.86	23.88 / 29.04	19.16 / 20.98
	EN-MS	16.59 / 7.98	0.76 / 0.34	4.82 / 1.26	14.51 / 2.31	8.16 / -4.92
Qwen2.5-3B	EN-ZH	38.32 / 36.44	2.72 / 2.80	10.99 / 11.20	44.42 / 41.06	21.43 / 21.29
	EN-TA	39.08 / 35.50	2.73 / 2.55	12.30 / 11.96	47.58 / 45.49	30.90 / 33.84
	EN-MS	34.00 / 14.60	2.11 / 1.03	5.97 / 3.56	33.91 / 17.70	16.09 / 10.70
Qwen2.5-7B	EN-ZH	40.89 / 39.07	11.95 / 12.20	59.88 / 60.92	217.69 / 208.36	234.54 / 239.63
	EN-TA	22.86 / 20.85	1.83 / 1.72	11.50 / 11.10	32.72 / 30.17	35.66 / 36.26
	EN-MS	9.71 / 2.06	0.74 / -0.10	6.27 / 3.64	12.16 / 1.82	11.38 / 5.36
SEALLM-7B	EN-ZH	50.45 / 51.94	11.09 / 11.21	60.22 / 60.69	170.47 / 160.94	163.13 / 142.51
	EN-TA	47.99 / 43.82	4.49 / 4.52	19.84 / 19.86	75.18 / 69.14	47.14 / 35.42
	EN-MS	34.20 / 17.03	3.29 / 2.53	12.01 / 27.78	47.49 / 4.22	13.08 / 27.78
Mistral-7B	EN-ZH	17.91 / 2.18	0.73 / -0.80	7.78 / 2.43	28.10 / 15.19	20.10 / 9.41
	EN-TA	-1.79 / -3.62	-2.20 / -2.85	3.90 / 1.90	8.93 / 6.68	13.76 / 11.95
	EN-MS	-11.88 / 15.12	-2.93 / 0.56	-1.21 / 2.93	-8.56 / 8.96	3.83 / -2.09
Ministral-8B	EN-ZH	9.92 / 13.16	1.15 / 0.69	9.58 / 9.85	30.37 / 34.98	29.41 / 33.61
	EN-TA	18.38 / 20.88	1.41 / 1.41	13.73 / 14.13	34.19 / 36.83	35.43 / 39.83
	EN-MS	-24.12 / 7.51	-2.73 / 0.63	-1.64 / 4.51	-8.07 / 9.89	7.89 / 2.76
LLaMA-3-8B	EN-ZH	23.51 / 25.21	3.14 / 3.12	12.88 / 12.79	29.85 / 34.26	19.70 / 29.68
	EN-TA	51.45 / 47.07	4.32 / 4.00	18.91 / 18.28	66.19 / 60.27	39.72 / 42.35
	EN-MS	46.81 / 21.73	4.85 / 3.27	22.19 / 16.94	50.56 / 23.01	39.34 / 17.83
Gemma-2-9B	EN-ZH	-1.45 / 10.66	0.12 / 0.82	4.10 / 2.32	8.56 / 11.61	1.68 / 7.95
	EN-TA	6.54 / 19.37	0.63 / 1.56	7.14 / 9.12	18.12 / 27.89	7.28 / 13.01
	EN-MS	7.81 / -6.27	0.71 / -0.61	3.60 / -1.41	8.48 / -3.45	5.31 / -7.06
SEA-Lion-Gemma-9B	EN-ZH	-22.93 / -6.64	-3.10 / -0.71	-3.27 / 1.73	-4.13 / 8.67	6.29 / 14.39
	EN-TA	-21.48 / 3.59	-2.61 / 0.24	1.74 / 7.08	-3.49 / 20.38	7.38 / 17.21
	EN-MS	-21.98 / -8.08	-2.95 / -0.99	-3.53 / -0.77	-8.16 / 3.18	9.09 / 3.03

TABLE 4.5: LoRA / QLoRA % Improvement over Few-Shot on CS-Sum (EN-TA)

objective of applying these PEFT methods was to assess whether synthetic CS training data generated via Gemini-2 could mitigate summarization errors, particularly semantic distortions and information omissions (see Section 4.3).

Significant and consistent improvements are observed primarily in the EN-ZH pair. Models like SEALLM-7B and Qwen2.5-7B achieve substantial gains (e.g., over 200% in METEOR), indicating that fine-tuning is highly effective when the target language pair is well-represented in the pretraining corpus.

For the EN-TA pair, moderate yet reliable improvements are observed across models. SEALLM-7B achieves notable improvements (47.99% LoRA, 43.82% QLoRA in ROUGE-L), while Qwen2.5-3B and LLaMA-3-8B also demonstrate steady gains. However, the variability across metrics suggests that model architecture and pre-training data significantly influence fine-tuning efficacy.

The EN-MS language pair exhibits the most inconsistent performance. While some models benefit from QLoRA, others, such as Mistral-7B and SEA-Lion-Gemma-9B, show performance regressions in certain metrics (ROUGE-L, BERTScore).

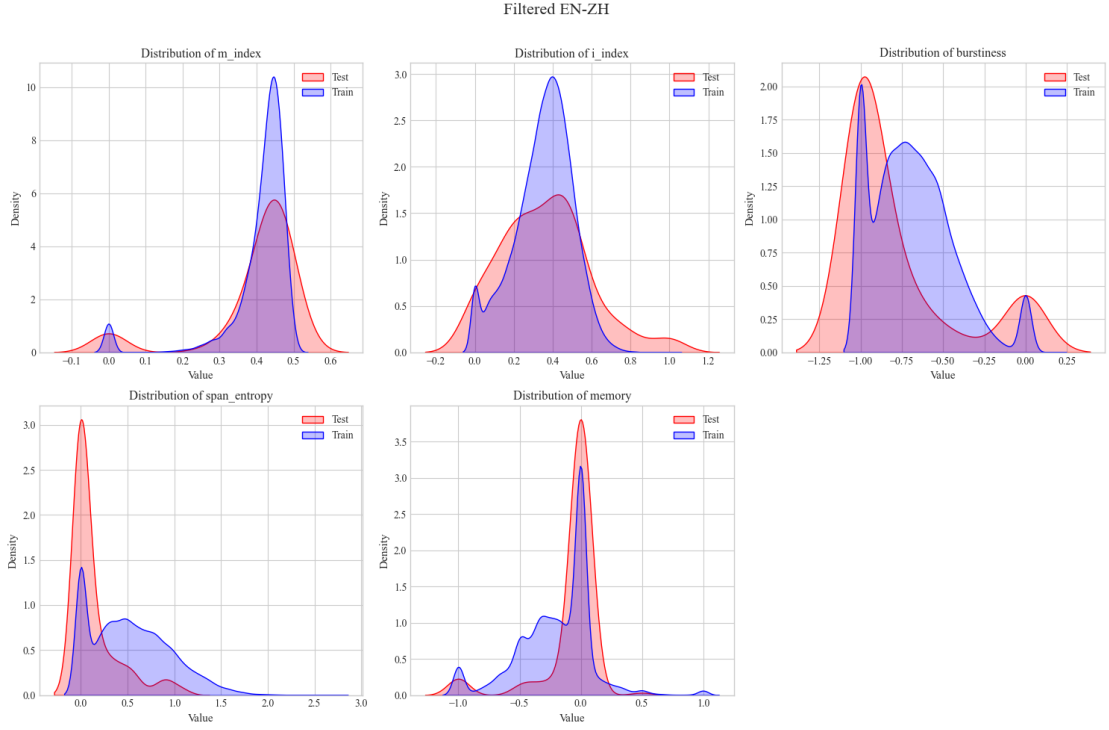


FIGURE 4.3: Distribution b/w filtered CS-Sum-Syn and CS-Sum for EN-ZH

The instability of results suggests that fine-tuning alone does not fully resolve the challenges of summarizing code-switched dialogues in lower-resource languages.

Despite these improvements, fine-tuning does not completely eliminate summarization errors. Semantic distortions and information omissions persist, particularly in EN-TA and EN-MS, reinforcing the need for more effective CS-aware pretraining strategies. The findings indicate that LoRA/QLoRA fine-tuning is most beneficial when models already have a strong multilingual foundation, with the highest impact observed in high-resource language pairs (EN-ZH), moderate but stable improvements in EN-TA, and inconsistent outcomes in EN-MS.

4.2.5 Training on filtered data

The difference in the distribution of CS-Sum and CS-Sum-Syn for EN-ZH (see Figure 4.1a) is much more than the differences in distribution for EN-MS and EN-TA. We hypothesize that this might be due to the tokenization strategy of the LLMs which is more robust for Roman script compared to Chinese script. The

% improvement gains for EN-ZH after LoRA and QLoRA finetuning on EN-ZH is also less compared to the other two CS pairs (Table 4.5).

To check if filtering CS-Sum-Syn to match CS-Sum’s distribution would improve the finetuning results, we treat the CS metrics shown in Table 4.1 as a multivariate gaussian. We calculated the mean $\boldsymbol{\mu}$ and the covariance matrix Σ of CS-Sum and used the Mahalanobis distance (eqn. 4.1) to calculate the distance between the instances in the training set and the test set.

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})} \quad (4.1)$$

We filtered the training data aggressively by selecting only the top 20 percentile of the training instances with the shortest distances. The filtered dataset had 3801 instances. The resulting distribution is shown in Figure 4.3 which matches CS-Sum’s distribution better than the unfiltered one (see Figure 4.1a).

We finetuned (QLoRA) all the 9 open-source LLMs on the filtered data for 3 epochs (4 for the unfiltered one) with all the other parameters staying the same. Figure 4.4 shows the % improvement of LLMs finetuned on filtered data over the LLMs finetuned on the unfiltered one.

It can be seen that only 2 LLMs - Mistral 7B and SEA-Lion-Gemma-9B show actual improvement while other LLMs either don’t improve much or have worse performances. Qwen2.5-7B suffers the worst when finetuned on filtered data. This suggests that filtering using Mahalanobis distance may only be beneficial for certain architectures while being detrimental to others. Further investigation is required to determine if alternative filtering thresholds or strategies could yield better generalization across LLMs

4.3 Error analysis

In this section, we present the three major errors that LLMs exhibit when summarizing CS dialogues. Our qualitative analysis of summaries generated by different LLMs across the three CS language pairs revealed the following common failure patterns:

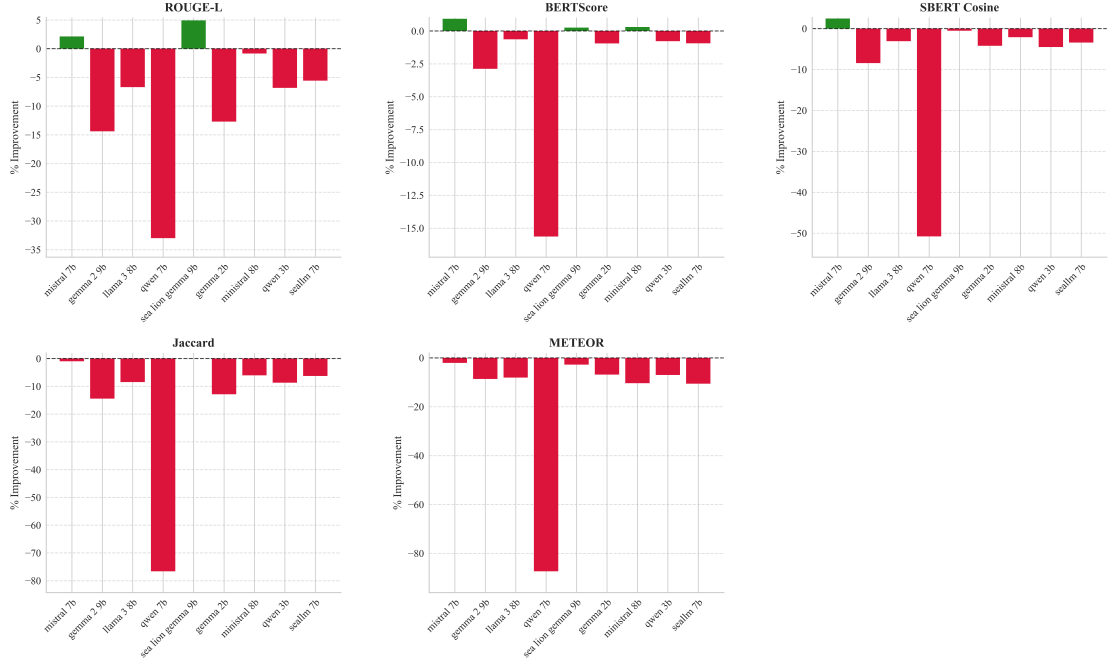


FIGURE 4.4: Filtered data % improvement over CS-Sum-Syn for EN-ZH

- **Code-Switching Loss (CSL):** The summary is generated only from the English parts of the dialogue, ignoring or missing key information from the non-English segments.
- **Meaning Shift from Poor Translation (MST):** The LLM misunderstands the code-switched text, leading to a summary that deviates from the true meaning of the dialogue.
- **Speaker Misattribution (SMA):** The summary assigns dialogue statements to the wrong speaker, changing the intended meaning or misrepresenting opinions.

Figure 4.5 shows an example of the CSL category, where the summary mainly focusses on the English parts and also wrongly says that Pam’s manager will call, while the CS dialogue mentions that Pam will make the call. For more examples of different categories see tables 4.7, 4.8 and 4.9. To find subtle errors like the ones in Figure 4.5, instead of employing annotators we used GPT-4o to identify the errors in the summaries.

Dialogue:
Lauren: Hi kau masih perlukan aku tak esok
Pam: Yes please!!
Lauren: Kau ada lagi rota?
Pam: No, tapi Manager aku balik esok so she may do some more then. **Aku akan call kau in the morning and let you know.**
Lauren: ok that's great
Pam: Did you have a good holiday?
Lauren: Yes, esok aku akan share ngan kau
Pam: Look forward to it!

Qwen2.5-7B summary:
Pam needs Lauren's help tomorrow. **Lauren will be contacted by Pam's manager** in the morning to confirm the arrangements. Pam asks Lauren if her holiday was good, and Lauren confirms it was, and she will share the details with Pam tomorrow.

FIGURE 4.5: CSL error example

Model	Lang	CSL		MST		SMA	
		Few-Shot	LoRA	Few-Shot	LoRA	Few-Shot	LoRA
Qwen2.5-7B	EN-ZH	71.64	81.75	32.84	48.91	9.70	26.28
	EN-TA	91.81	79.23	76.32	58.64	57.31	32.58
	EN-MS	78.95	82.05	35.96	42.74	16.67	24.79
LLaMA-3-8B	EN-ZH	74.63	71.17	35.82	45.07	16.41	21.35
	EN-TA	90.94	56.92	80.70	37.89	55.85	19.93
	EN-MS	73.68	62.91	40.35	33.38	14.04	17.33
Gemma-2-9B	EN-ZH	59.70	85.40	18.66	57.66	8.21	35.77
	EN-TA	71.35	82.75	35.09	67.84	22.22	46.20
	EN-MS	59.65	79.60	18.42	52.53	8.77	29.45
SEA-Lion-Gemma-9B	EN-ZH	53.73	83.94	11.19	76.64	3.73	52.55
	EN-TA	60.53	84.50	27.78	76.61	17.54	49.71
	EN-MS	48.25	86.32	14.04	64.96	7.02	46.15

TABLE 4.6: Comparison of error rates (%) in Few-Shot vs. LoRA fine-tuned models across different CS language pairs.

To find how common the three errors are, in the summaries generated by different LLMs across different languages and to answer if finetuning with synthetic data mitigates these issues, we calculated the error rate for the three category of errors, for Qwen2.5-7B, LLaMA-3-8B, Gemma-2-9B and SEA-Lion-Gemma-8B since the first two LLMs showed the most % improvement among the 9 LLMs after LoRA finetuning on CS-Sum-Syn (Table 4.5) and latter two because of their strong Few-Shot performance (Table 4.4)

Table 4.6 shows the error rate comparison between the Few-Shot and the LoRA settings. The error rate on EN-ZH increases for all the LLMs after finetuning on the synthetic data which is expected because of the difference in distribution between CS-Sum and CS-Sum-Syn (see Figure 4.1a and Table 4.2). All the models’ error rates on EN-TA are generally higher than the other CS languages. Though finetuning with synthetic data helps a little for Qwen and LLaMA, the scores are still higher than the Few-Shot settings for other CS languages.

The high CSL error rates for the LLMs indicate that, more often than not LLMs rely on the English part of the text which is a serious problem, since important things like emotions, phrases difficult to say in English for a person whose mother tongue is not English are usually not conveyed in English. Moderate error rates for SMA and MST also highlight a serious issue, since attributing a person to something they didn’t say or think and wrongly interpreting the CS parts of the dialogues are most likely to change the entire meaning of the dialogue.

4.4 Summary

In this chapter, we introduced CS-Sum, the first CS dialogue-to-English summarization benchmark across three language pairs: EN-TA, EN-ZH, and EN-MS. Our experiments revealed that LLMs perform inconsistently across CS pairs, with some open-source models outperforming SEA-specific models, highlighting that CS data must be present in pretraining, not just added through fine-tuning. Furthermore, our analysis identified three major error types—CSL, MST, and SMA—persisting across all models, demonstrating that current LLMs lack intrinsic CS comprehension. The quantitative results reinforce the severity of these failures, showing that fine-tuning alone does not fully resolve CS-specific challenges.

EN Dialogue	CS Dialogue	LLM Summary	Error Explanation
<p>Matthew: I have a cold. Need y'all support. It's painful</p> <p>Lisa: Fight it. Drink orange juice. Make some soup</p> <p>Matthew: I'll try</p> <p>Lisa: I believe in u</p> <p>Matthew: Thank u. Apparently it's an allergy mixed up with virus</p> <p>Lisa: Ahh it sucks.</p> <p>Lisa: Take a rest from evrthg.</p> <p>Matthew: Ikr. Hitting the hay now</p> <p>Lisa: Good night 'Thew!</p>	<p>Matthew: Enakku cold irukku. Unga ellaroda support um vennum. Athu romba painful</p> <p>Lisa: Atha fight pannu. Orange juice kudi. Soup pannu</p> <p>Matthew: Naa try panren</p> <p>Lisa: Naa unna believe pan-ran</p> <p>Matthew: Thank u. Athu oru allergy and virus mix aana onnu.</p> <p>Lisa: Ahh athu kaduppu than.</p> <p>Lisa: Ellathula irunthum rest edu.</p> <p>Matthew: Athu enakku theriyum. Ipo poi padukka poren</p> <p>Lisa: Apo Good night 'Mathew!</p>	<p>Matthew is feeling unwell due to a cold and an allergy.</p> <p>Lisa suggests that he drink orange juice and soup to feel better.</p> <p>Matthew tries it and feels a bit better.</p>	<p>Matthew says that he will try the pill but the summary mentions that he had tried it and feels better</p>
<p>Anna: is anyone going to pick Mark from the airport?</p> <p>Marcus: i could but when and where from?</p> <p>Anna: Sydney, Thursday at 3</p> <p>Marcus: am or pm? :D</p> <p>Leslie: haha fortunately pm:D</p> <p>Marcus: hmm i have a meeting at 1. I don't think i can make it</p> <p>Leslie: well i guess it will take him some time after landing, reclaiming luggage etc</p> <p>Anna: yeah I reckon it's fine if you're there at 4</p> <p>Marcus: oh well ok then</p> <p>...</p>	<p>Anna: 有人去机场接Mark吗?</p> <p>Marcus: 我可以, 但是什么时候去哪里呢?</p> <p>Anna: Sydney, 星期四3点</p> <p>Marcus: am 还是pm? :D</p> <p>Leslie: haha, 幸运的是下午:D</p> <p>Marcus: hmm 我1点有个会议。我想我去不了了</p> <p>Leslie: well 我想after his landing 他需要一些时间, reclaiming luggage etc</p> <p>Anna: 是的, 我认为如果你4点到那里没问题了</p> <p>Marcus: oh 那好吧</p> <p>...</p>	<p>Marcus originally plans to pick up Mark at the airport but has a meeting at 1 PM and can't make it. Leslie suggests Marcus arrive after 4 PM to give Mark enough time to reclaim luggage. Anna will inform Mark about Marcus's new plan.</p>	<p>Anna was the one who suggest Marcus to go at 4 PM which is wrongly attributed to Leslie</p>
<p>Ken: You are an idiot.</p> <p>Greg: What?</p> <p>Ken: Yeah you are, I want my money back</p> <p>Greg: Was your account hacked by some prankster?</p> <p>Ken: No, I'm the prankster, just having a laugh at your expense</p> <p>Greg: Well, you are an idiot too then XD</p>	<p>Ken: 操你, 你pimp</p> <p>Greg: 什么?</p> <p>Ken: 操你, 我要回我的钱</p> <p>Greg: 你的账户被恶作剧者hacked了吗?</p> <p>Ken: 不, 我才是恶作剧者, 只是开个玩笑以你为代价</p> <p>Greg: Well, 那你也操XD</p>	<p>Ken and Greg were joking with each other. Ken played a prank on Greg but now he wants his money back. Greg was not happy about it</p>	<p>The dialogue is just a friendly banter and there's actually no transaction of money.</p>

TABLE 4.7: Error examples

EN Dialogue	CS Dialogue	LLM Summary	Error Explanation
<p>...</p> <p>Person2: Large deposits of coal were found nearly and so many industries located themselves here. The village quickly grew into a key industrial centre.</p> <p>Person1: As the city grew, it must have absorbed many village nearby.</p> <p>Person2: Yes, it did. The names of those village survive as the names of parts of the city.</p> <p>Person1: I see. Are there any building more than 200 years old in your city?</p> <p>Person2: Oh, yes. Several of the buildings from the villages still survive. Many of them were inns for travelers and today survive as pubs. There was a castle near one village, so our city has a castle too.</p> <p>Person1: Really? So your city does have some old history after all.</p>	<p>...</p> <p>Person2: 附近发现了大量coal。许多工业都设在这里。这个村庄迅速发展成为一个重要的industrial centre。</p> <p>Person1: 随着城市发展, 它一定absorbed了附近的许多村庄。</p> <p>Person2: 是的。这些村庄的名字作为城市部分地区的名字而留存下来。</p> <p>Person1: I see. 你的城市里有超过200年历史的building吗?</p> <p>Person2: 村庄里的一些buildings仍然保存完好。其中许多都是旅客的inns, 如今作为pubs保留了下来。一个村庄附近有一座castle, 所以我们的城市也有一座castle。</p> <p>Person1: 真的吗? 原来你们城市也有悠久的历史啊。</p>	<p>Person2 tells Person1 how their city grew and has some buildings over 200 years old. Person2 also introduces a castle nearby</p>	<p>The summary is based on only the English words of the dialogue and also Person2 talks about a castle, does not introduce it</p>
<p>Alan: jfile_photo;</p> <p>Alan: look what I just found</p> <p>Robert: dude, that's just nasty and you know it</p> <p>Robert: it has no sugar, no taste, and additional cinnamon flavoring</p> <p>Alan: yeah, I know - that's awesome</p> <p>Robert: you sir have a very strange tastes :P</p> <p>Alan: well, and I found a perfect company for it jfile_photo;</p> <p>Robert: oh, that's more like it!</p> <p>Robert: but does the whiskey go well with the cinnamon? flavored whiskey is the worst...</p> <p>...</p>	<p>Alan: file.photo;</p> <p>Alan: tengok ape aku jumpe</p> <p>Robert: dude, that's just nasty and you know it</p> <p>Robert: it has no sugar, no taste, and additional cinnamon flavoring</p> <p>Alan: yeah, I know - itulah yang awesome</p> <p>Robert: you sir ada tastes pelik :P</p> <p>Alan: well, and I found a perfect company for it jfile_photo;</p> <p>Robert: oh, that's more like it!</p> <p>Robert: Tapi does the whiskey go well with the cinnamon? flavored whiskey rabak...</p> <p>...</p>	<p>Alan shares a photo of a cinnamon-flavored whiskey with Robert, who initially finds it disgusting. Alan defends the unique taste and shares another photo of him enjoying it with a companion. Robert jokingly points out Alan's unusual taste preferences but admits the cinnamon taste isn't overpowering when paired with enough whiskey. ...</p>	<p>The summary wrongly states that Alan has a companion when he was actually referring to eating cinnamon along with it</p>

TABLE 4.8: Error examples

EN Dialogue	CS Dialogue	LLM Summary	Error Explanation
<p>Callan: Something's wrong with my Samsung S8.</p> <p>Wade: What, exactly?</p> <p>Callan: All I get is a black screen.</p> <p>Wade: Did you drop it or something?</p> <p>Callan: I think I know, this has happened before.</p> <p>Wade: It did? What happened before?</p> <p>Callan: It overheats and then it takes a while for it to restart again..</p> <p>Wade: That sucks.. Be carefull with the battery, it can be dangerous.</p> <p>...</p>	<p>Callan: Samsung S8 aku ada problem.</p> <p>Wade: What, exactly?</p> <p>Callan: Aku asyik dapat black screen.</p> <p>Wade: Kau terjatuhkan ke apa?</p> <p>Callan: Aku rasa aku tahu kot, ni pernah happened kat aku.</p> <p>Wade: It did? What happened before?</p> <p>Callan: Dia overheat pastu it takes a while for it to restart again..</p> <p>Wade: That sucks.. Be carefull with the battery, boleh jadi bahaya.</p> <p>...</p>	<p>The speaker, Callan, is having issues with his Samsung S8 as it keeps getting black screens. Wade suggests it might be overheating. Callan intends to take it to a store as it's still under warranty</p>	<p>Callan was the one who suggested the phone might be overheating</p>
<p>Person1: What do you know about our company?</p> <p>Person2: Well, as I know this company is one of the largest and best in this field of business. It mainly produces toys for children. It employs more than 10,000 people throughout the world. The president now is Mr. Jackson. The Shanghai branch was founded five years ago with a staff of more than 2,000.</p> <p>Person1: Very good. If you enter this company, what department would you like to work in?</p> <p>Person2: May I work in sales department?</p> <p>Person1: OK. Oh, how much do you make at your current job, if you don't mind my asking?</p> <p>...</p>	<p>Person1: Enga company pathi enna therium ungaluku?</p> <p>Person2: Indha field indha company dhaan largest and best nu therium. Main focus vandhu children toys produce panradhu. Current president Mr. Jackson. Shanghai la 5 year munnadi oru branch start panirkeenga. Anga 2000 employees irukanga. Total ah 10000 employees.</p> <p>Person1: Very good. Inga job kedaicha, endha department la work pananum ungaluku?</p> <p>Person2: Sales department la?</p> <p>Person1: OK. Current job la evlo earn panreenga? If you don't mind my asking?</p> <p>Person2: Current salary 2000 yuan plus konjam allowances and bonus ku apram 3000 yuan per month. Neenga evlo pay panuveenga?</p> <p>Person1: Clerks ku starting salary 2500 yuan. 3 months ku apram raise varum.</p> <p>Person2: Fringe benefits edhavadhu iruka?</p> <p>...</p>	<p>Person2 is describing their current company which is the largest and best in the field, focusing on children's toys. They have a branch in Shanghai with 2000 employees, totaling 10,000 employees worldwide. Person1 is interested in a sales position and asks about the salary and benefits, learning that Person2 currently earns 3000 yuan per month with allowances and bonuses. Person1 explains that the starting salary for clerks is 2500 yuan, ...</p>	<p>LLM has completely switched Person1 and Person2 in the first half of the summary</p>

TABLE 4.9: Error examples

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this work, we introduced DiaSynth, a synthetic dialogue generation framework designed to produce high-quality, contextually rich dialogues across various domains. Our experiments demonstrated that models fine-tuned on DiaSynth-generated data significantly outperform their base models, as evidenced by substantial improvements in summarization metrics. These results highlight the potential of DiaSynth as an effective tool for generating dialogue data, particularly for domains where training data is scarce. Additionally, we found that different LLMs excel in different dialogue structures, with LLaMA-3 performing better for informal dialogues and GPT-4o for more structured settings. This insight suggests that leveraging open-source LLMs may be more advantageous for generating human-like conversational data. Despite certain limitations, such as varying LLM performance across dialogue types and knowledge gaps in zero-shot generation, DiaSynth offers a promising approach to dialogue data generation and provides a valuable resource for advancing dialogue systems.

In parallel, we introduced CS-Sum, the first benchmark for CS dialogue-to-English summarization across three language pairs: EN-TA, EN-ZH, and EN-MS. Our experiments showed that LLMs perform inconsistently across CS pairs, with open-source models outperforming SEA-specific models, emphasizing the importance of including CS data in pretraining rather than just adding it through fine-tuning. Moreover, our analysis identified three major error types—CSL, MST, and SMA

— which persist across all models, indicating that current LLMs lack intrinsic CS comprehension. The quantitative results further highlight the severity of these issues, showing that fine-tuning alone does not fully resolve the challenges posed by CS data. By releasing CS-Sum, along with our codebase and evaluation framework, we aim to drive future research in multilingual NLP, providing a benchmark to better understand and address the shortcomings of LLMs when processing code-switched data.

5.2 Future Work

Despite the promising results, our approach with DiaSynth has some limitations that suggest potential directions for future research. One key area for improvement is the performance variability across different dialogue structures. For instance, certain models like LLaMA-3 perform better for informal dialogues, such as those found in the SAMSum dataset, while others like GPT-4o excel in more structured, formal dialogues, such as DialogSum. This indicates that there is no single model that can universally handle all types of dialogue structures. Future work could focus on combining models or selecting the best model for each type of dialogue to improve overall performance. Additionally, DiaSynth’s reliance on zero-shot generation for personas and sub-topics may result in less coherent or less accurate persona simulations, particularly in domains where models are not fine-tuned for specific contexts. To address these challenges, future research should explore incorporating more domain-specific training and fine-tuning to improve the accuracy and coherence of persona simulations. Furthermore, DiaSynth’s reliance on LLMs introduces challenges related to hallucinations and the compute requirements of these models. While our hallucination study shows that the rates are acceptable, further research should aim to reduce hallucinations and explore ways to optimize the computational efficiency of the framework.

Regarding CS-Sum, our results indicate that LLMs still struggle with understanding code-switching, even after fine-tuning. A key area for future work would be training models on real-world code-switched data instead of relying solely on synthetic examples. This would allow the models to better capture the complexities and nuances of natural code-switching in multilingual contexts. Another avenue for improvement would be exploring better tokenization methods to handle

mixed-language inputs more effectively, which is a significant challenge in current models. Since performance varies across different language pairs, expanding CS benchmarks to include more language combinations—particularly low-resource languages—would be highly beneficial. Additionally, we observed that models often favor English over non-English parts of code-switched text, indicating a bias in their training. Future research should focus on developing better training techniques to reduce this bias and enhance the models’ ability to process and understand non-English components more equally. Addressing these challenges will be crucial in creating LLMs that truly understand and generate accurate code-switched content across diverse languages.

Bibliography

- [1] Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. doc2dial: A goal-oriented document-grounded dialogue dataset. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.652. URL <https://aclanthology.org/2020.emnlp-main.652>. 1, 9
- [2] Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. MedDialog: Large-scale medical dialogue datasets. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.743. URL <https://aclanthology.org/2020.emnlp-main.743>. 7
- [3] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL <https://aclanthology.org/D18-1547>. 1, 9
- [4] Ji-Eun Han, Jun-Seok Koh, Hyeon-Tae Seo, Du-Seong Chang, and Kyung-Ah Sohn. PSYDIAL: Personality-based synthetic dialogue generation using large language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13321–13331, Torino, Italia, May 2024. ELRA and ICCL. URL <https://aclanthology.org/2024.lrec-main.1166>. 2, 8
- [5] Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024. 2, 8

- [6] Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. Evaluating code-switching translation with large language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6381–6394, Torino, Italia, May 2024. ELRA and ICCL. URL <https://aclanthology.org/2024.lrec-main.565/>. 2, 10
- [7] Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri Aji. Multilingual large language models are not (yet) code-switchers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.774. URL <https://aclanthology.org/2023.emnlp-main.774/>. 2, 9, 10
- [8] Zheng Xin Yong, Ruochen Zhang, Jessica Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. Prompting multilingual large language models to generate code-mixed texts: The case of south East Asian languages. In Genta Winata, Sudipta Kar, Marina Zhukova, Thamar Solorio, Mona Diab, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali, editors, *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63, Singapore, December 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.calcs-1.5/>. 2, 9
- [9] Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. LinCE: A centralized benchmark for linguistic code-switching evaluation. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.223/>. 3, 10
- [10] Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. GLUECoS: An evaluation benchmark for code-switched NLP. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.329. URL <https://aclanthology.org/2020.acl-main.329/>. 3, 10
- [11] Laiba Mehnaz, Debanjan Mahata, Rakesh Gosangi, Uma Sushmitha Gunturi, Riya Jain, Gauri Gupta, Amardeep Kumar, Isabelle G. Lee, Anish

- Acharya, and Rajiv Ratn Shah. GupShup: Summarizing open-domain code-switched conversations. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6177–6192, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.499. URL <https://aclanthology.org/2021.emnlp-main.499/>. 3, 11
- [12] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.449. URL <https://aclanthology.org/2021.findings-acl.449>. 7, 20
- [13] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAM-Sum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL <https://aclanthology.org/D19-5409>. 7, 20
- [14] G Richard Tucker. A global perspective on bilingualism and bilingual education. *Georgetown University Round table on Languages and Linguistics 1999*, 01 2001. 7
- [15] Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. Do multilingual users prefer chat-bots that code-mix? let’s nudge and find out! *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1), May 2020. doi: 10.1145/3392846. URL <https://doi.org/10.1145/3392846>. 7
- [16] Shana Poplack. *Code Switching: Linguistic*, pages 2062–2065. 12 2001. ISBN 9780080430768. doi: 10.1016/B0-08-043076-7/03031-X. 7
- [17] Sebastian Steindl, Ulrich Schäfer, and Bernd Ludwig. Generating synthetic dialogues from prompts to improve task-oriented dialogue systems. In *KI 2023: Advances in Artificial Intelligence: 46th German Conference on AI, Berlin, Germany, September 26–29, 2023, Proceedings*, page 207–214, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-42607-0. doi: 10.1007/978-3-031-42608-7_17. URL https://doi.org/10.1007/978-3-031-42608-7_17. 8
- [18] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 8

- [19] Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Tamar Solorio. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.185. URL <https://aclanthology.org/2023.findings-acl.185/>. 10
- [20] Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Peng Xu, Yan Xu, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J. Barezi, Qifeng Chen, Xiaojuan Ma, Bertram Shi, and Pascale Fung. ASCEND: A spontaneous Chinese-English dataset for code-switching in multi-turn conversation. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7259–7268, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.788/>. 11
- [21] Ruochen Zhang and Carsten Eickhoff. CroCoSum: A benchmark dataset for cross-lingual code-switched summarization. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4113–4126, Torino, Italia, May 2024. ELRA and ICCL. URL <https://aclanthology.org/2024.lrec-main.367/>. 11
- [22] Nathaniel Oco and Rachel Edita Roxas. Pattern matching refinements to dictionary-based code-switching point detection. In Ruli Manurung and Francis Bond, editors, *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 229–236, Bali, Indonesia, November 2012. Faculty of Computer Science, Universitas Indonesia. URL <https://aclanthology.org/Y12-1024/>. 11
- [23] Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. Normalization of Indonesian-English code-mixed Twitter data. In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5554. URL <https://aclanthology.org/D19-5554/>.
- [24] Jiaan Wang, Fandong Meng, Ziyao Lu, Duo Zheng, Zhixu Li, Jianfeng Qu, and Jie Zhou. ClidSum: A benchmark dataset for cross-lingual dialogue summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7729, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi:

- 10.18653/v1/2022.emnlp-main.526. URL <https://aclanthology.org/2022.emnlp-main.526/>. 11
- [25] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088. 13
- [26] Shikib Mehri and Maxine Eskenazi. Unsupervised evaluation of interactive dialog with DialoGPT. In Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes, editors, *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 1st virtual meeting, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.sigdial-1.28. URL <https://aclanthology.org/2020.sigdial-1.28>. 17
- [27] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.365. URL <https://aclanthology.org/2024.naacl-long.365>. 17
- [28] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chengguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL <https://aclanthology.org/2023.emnlp-main.153>. 17
- [29] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT : Large-scale generative pre-training for conversational response generation. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.30. URL <https://aclanthology.org/2020.acl-demos.30/>. 17
- [30] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 20
- [31] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiao wen Dong, Haodong Duan,

- Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhen Jin, Zhikai Lei, Jiaying Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kui-Jie Liu, Xiaoran Liu, Chen Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xing Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Rui Ze Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fen-Fang Zhou, Zaida Zhou, Jingming Zhuo, Yi-Ling Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report. *ArXiv*, abs/2403.17297, 2024. URL <https://api.semanticscholar.org/CorpusID:268691939>. 20
- [32] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 20
- [33] Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.450. URL <https://aclanthology.org/2020.acl-main.450>. 20
- [34] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. 20, 38, 42
- [35] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>. 20, 38, 42
- [36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423/>. 21

- [37] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL <https://aclanthology.org/2020.acl-main.703>. 22
- [38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 22
- [39] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 22
- [40] Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL <https://aclanthology.org/2023.emnlp-main.557>. 31
- [41] Robert Friel and Atindriyo Sanyal. Chainpoll: A high efficacy method for llm hallucination detection. *ArXiv*, abs/2310.18344, 2023. URL <https://api.semanticscholar.org/CorpusID:264590664>. 31
- [42] Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E. Bullock, and Almeida Jacqueline Toribio. Metrics for modeling code-switching across corpora. In *Interspeech 2017*, pages 67–71, 2017. doi: 10.21437/Interspeech.2017-1429. 34, 35
- [43] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 35
- [44] M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. ISSN 0016-0032. doi: [https://doi.org/10.1016/S0016-0032\(96\)00063-4](https://doi.org/10.1016/S0016-0032(96)00063-4). URL <https://www.sciencedirect.com/science/article/pii/S0016003296000634>. 37
- [45] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 37

- [46] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 37
- [47] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L’elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL <https://api.semanticscholar.org/CorpusID:263830494>. 37
- [48] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 37, 43
- [49] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 37
- [50] Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. SeaLLMs - large language models for Southeast Asia. In Yixin Cao, Yang Feng, and Deyi Xiong, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 294–304, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.28. URL <https://aclanthology.org/2024.acl-demos.28/>. 37
- [51] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 37
- [52] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410/>. 38, 42
- [53] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of*

- the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL <https://aclanthology.org/W05-0909/>. 38, 40, 42
- [54] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>. 41
- [55] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA, 2023. Curran Associates Inc. 41