# Evaluating Code-Switching Translation with Large Language Models

**Muhammad Huzaifah**[*1], **Weihua Zheng**[*1], **Nattapol Chanpaisit**[†2], **Kui Wu**[1]
[1]Institute for Infocomm Research (I$^2$R), Agency for Science, Technology and Research, Singapore
[2]Nanyang Technological University, Singapore
{huzaifah_md_shahrin, zheng_weihua}@i2r.a-star.edu.sg

**Abstract**

Recent advances in large language models (LLMs) have shown they can match or surpass finetuned models on many natural language processing tasks. Currently, more studies are being carried out to assess whether this performance carries over across different languages. In this paper, we present a thorough evaluation of LLMs for the less well-researched code-switching translation setting, where inputs include a mixture of different languages. We benchmark the performance of six state-of-the-art LLMs across seven datasets, with GPT-4 and GPT-3.5 displaying strong ability relative to supervised translation models and commercial engines. GPT-4 was also found to be particularly robust against different code-switching conditions. Several methods to further improve code-switching translation are proposed including leveraging in-context learning and pivot translation. Through our code-switching experiments, we argue that LLMs show promising ability for cross-lingual understanding.

**Keywords:** code-switch, machine translation, large language models, evaluation, prompt engineering

## 1. Introduction

Code-switching, that is the alternation of multiple languages in an utterance (Poplack, 1978), is a common phenomenon arising in multilingual communities. Such language use is also prevalent in online discourse, especially under the informal context of social media. With the increasing interconnectedness of our world, effective translation systems for code-switching are of growing importance. Nevertheless, while advances in neural machine translation (NMT) have led to significant leaps in translation ability, the code-switch setting remains a considerable challenge (Winata et al., 2023).

Historically, NMT systems have struggled with code-switching because they were typically designed for monolingual text, where the model learns an alignment between monolingual source and target data through cross-attention. Consequently, such models are brittle to inputs containing multiple languages during inference. NMT training is moreover still largely reliant on huge amounts of parallel data, which is relatively scarce for code-switched text. More advanced multilingual models have been proposed but their effectiveness has not proven to be definitive (Winata et al., 2021).

Recent breakthroughs in decoder-based large language models (LLMs) have revolutionised the field of natural language processing (NLP). LLMs have been shown to not only improve performance across a wide variety of NLP problems (Gao et al., 2021) but also provide a common natural language interface to interact with the model. As opposed to traditional NMT systems, LLMs are trained for language modelling for which parallel data is unnecessary, and are therefore potentially better suited to handle translation of code-switched text.

Various studies have evaluated the performance of LLMs for the translation task (Jiao et al., 2023; Hendy et al., 2023), including research on more effective prompting techniques (Vilar et al., 2023; Zhang et al., 2023a), and for specific scenarios like document-level (Wang et al., 2023) and multilingual translation (Zhu et al., 2023). There has been mixed results on the use of LLMs for general translation, with some reporting competitive ability on high-resource languages but lagging behind other supervised NMT models especially on lower-resource languages. Notwithstanding, we observe a clear trend of LLMs getting better with each iteration, notably with the release of GPT-3.5 (Brown et al., 2020), and subsequently GPT-4 (OpenAI, 2023). With regards to code-switching, Zhang et al. (2023b) argued that multilingual LLMs were not necessarily compatible with such inputs on a variety of tasks. However, we note that their analysis was mostly carried out on the previous generation of LLMs, and they found GPT-3.5 to be much more comparable to finetuned models. Similarly, Yong et al. (2023) found that ChatGPT (GPT-3.5) outperformed other multilingual LLMs in generating code-switched texts for several South East Asian languages.

In this work, we focus on assessing the translation ability of state-of-the-art LLMs for code-

[*] Equal contribution
[†] Work carried out during intership at I$^2$R



*LREC-COLING 2024*, pages 6381–6394
20-25 May, 2024. 

switched text relative to supervised NMT models and commercial engines. We observe that GPT-4 and GPT-3.5 are able to handle code-switching inputs very well and may be considered on par or better than the commercial engines, while the other LLMs are far more inconsistent. Further experiments reveal that GPT-4 is significantly more robust than Google Translate against heavier code-switching and displays evidence of cross-lingual understanding given different code-switching distributions. We forward different methods to enhance translation ability in GPT-4, firstly via in-context learning for which we propose a new selection strategy CMS, and secondly through pivot translation into English and the matrix language. We also publicly make available a code-base and new synthetic code-switching datasets derived from Flores-200[1].

## 2. Experimental Setup

### 2.1. Large language models

LLMs belong to a class of decoder architectures that learn to autoregressively generate the next token (commonly a subword) given previous tokens, following a so-called language modelling objective. Compared to prior language models, they are characterised by their large parameter size containing billions to a trillion weights and vast amounts of training data amounting to trillions of tokens. With further instruction finetuning, LLMs have displayed an uncanny ability to adhere to human-generated prompts.

While there has been many flavours of LLMs released recently, we chose several that are well benchmarked to provide a comprehensive representation of the current state-of-the-art. Given that prior work has shown that performance scales with parameter size (Kaplan et al., 2020), we only evaluated the biggest available version for each model. Furthermore, we opted for the instruction finetuned versions of the models to maintain a consistent way of prompting them. Additional generation parameters that may be exposed through the API, such as temperature, were left at their defaults. For models hosted online, in particular GPT-4 and Bard (as well as the commercial MT engine baselines), there is a possibility of updates to the underlying model over time, which may alter its behaviour. For full disclosure, we accessed these models in the period between June-September 2023 for this study. We consider the following LLMs in our evaluation:

**GPT-4** (OpenAI, 2023) is the latest LLM offered by OpenAI. It has displayed remarkable capabilities in zero-shot and few-shot scenarios, including for general translation (Jiao et al., 2023). Unfortunately, OpenAI has not disclosed details on their model and training strategies, which has hampered the reproducibility of their methodology. We directly accessed GPT-4 though the ChatGPT web interface[2] via its Plus subscription service.

**GPT-3.5** (Brown et al., 2020) is a 175B parameter model powering ChatGPT before the roll-out of GPT-4, and was mostly responsible for the explosion of interest in such applications by the general public. At that time it was the leading LLM in many NLP benchmarks before being superseded by the more advanced GPT-4, especially on more complex reasoning tasks. Our evaluation is conducted on the gpt-3.5-turbo-0613 version through the chat completions API.

**Bard** is a conversational chatbot released by Google in a similar vein to ChatGPT. Its latest iteration is based on the PaLM-2 model (Anil et al., 2023), although further technical details were not made public. Having been trained on a large corpora of multilingual text, PaLM-2 is claimed to excel at multilingual tasks including translation, for which it outperformed Google Translate and PaLM on the WMT21 test set. We accessed Bard through the unofficial Bard-API package[3] that pulls responses from Bard[4] through cookies.

**LLaMA-2** (Touvron et al., 2023b) is a family of LLMs released by Meta with parameter sizes ranging from 7B to 70B, and is the successor to the popular LLaMA (Touvron et al., 2023a). The latter was one of the first open-source LLMs trained at scale and so was well adopted by the research community. LLaMA-2 further improved performance having been trained on 40% more data and twice the context length. We adopted the official 70B instruction-finetuned version available on Huggingface[5].

**Falcon** (Almazrouei et al., 2023) is an LLM developed by the Technology Innovation Institute (TII) and was the best performing LLM on the Open LLM leaderboard for a period of time, surpassing the original LLaMA model. It was mostly trained on the open-source RefineWeb dataset

[1] https://github.com/muhdhuz/CodeSwitch_Text_Generator

[2] https://chat.openai.com/

[3] https://github.com/dsdanielpark/Bard-API

[4] https://bard.google.com/

[5] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

(Penedo et al., 2023), which was curated through innovative filtering techniques. Here, the Falcon-40B-instruct[6] variant was used. We note that TII has more recently released a 180B parameter version that shows better performance compared to its predecessors.

**Phoenix** (Chen et al., 2023) is an LLM that focuses on multilingual performance, in particular for Chinese and other non-Latin languages, for which it was shown to outperform other open-source models. Phoneix uses BLOOMZ (Muennighoff et al., 2023) as a base model that is further fine-tuned on multilingual instruction and conversation data. We utilised Phoenix-inst-chat-7b, available on Github[7].

## 2.2. Baselines

We compare the above LLMs against systems commonly used for translation, namely the commercial MT engines Google Translate[8] and DeepL Translate[9], and the massive multilingual translation model NLLB (NLLB-Team et al., 2022), the largest of which is comparable in size to the LLMs, but is instead trained in a supervised fashion. We consider the nllb-moe-54B[10] and the nllb-200-distilled-1.3B[11] variants, available on Huggingface. Since the traditional MT systems were not built specifically for code-switching, we were limited to specifying only a single language as input. In this case, the matrix language (Myers-Scotton, 1993), that is the language that occurs with the highest frequency within the code-switch, was chosen as the source language[12]. The matrix language can also be referred to as the dominant language, with the minor language being the embedded language.

## 2.3. Data

There is a dearth of high quality parallel code-switching data in the wild containing both code-switch text and their translations. We take advantage of prior limited efforts to derive such datasets from speech or social media data where code-switching is most common, and supplement them with synthetic data generated from Flores-200 (NLLB-Team et al., 2022). For all evaluation we only consider a code-switching source made up of two languages to a single target language.

### 2.3.1. Real data

The three open-source datasets below were chosen for this evaluation. All datasets were first pre-processed by deduplication, removing empty lines, and removing lines where source and target are identical.

- Hindi-English (HI-EN) to English from the LinCE code-switching benchmark (Aguilar et al., 2020). The development set is utilised, containing 892 lines.
- Spanish-English (SP-EN) to English from the Bangor Miami speech dataset (Deuchar et al., 2014). We followed Weller et al. (2022) in preparing the data, with a final total of 3204 lines for evaluation.
- Indonesian-English (ID-EN) to Indonesian derived from Twitter/X posts (Barik et al., 2019). This dataset was not ideal since the source matrix language coincides with the target language so there are many overlaps between the two. We used 815 lines for evaluation.

### 2.3.2. Synthetic data

For more in-depth investigation of code-switching properties and better coverage of diverse language pairs, we constructed pseudo-code-switching data from the multilingual translation dataset Flores-200 (NLLB-Team et al., 2022). Flores-200 contains parallel text in over 200 languages, including very low-resource ones, sourced primarily from the Wikimedia project and translated by professional human translators. The parallel data over a huge number of languages provides much flexibility in choosing suitable languages for mixing.

The synthetic code-switching data generation pipeline is primarily based on the GCM toolkit (Rizvi et al., 2021). We utilise their implementation of the Matrix Language theory (Myers-Scotton, 1993) to generate valid code-switching text from parallel data in two specified languages. The original pipeline contains three major stages, namely a word-level alignment stage between input sentences, an analysis stage with sentence parsing, and a generation stage where the data from the prior stages are combined with linguistic theory to decide on word substitutions. We enhanced several aspects of the pipeline, including:

[6]https://huggingface.co/tiiuae/falcon-40b-instruct
[7]https://github.com/FreedomIntelligence/LLMZoo
[8]https://translate.google.com/
[9]https://www.deepl.com/translator
[10]https://huggingface.co/facebook/nllb-moe-54b
[11]https://huggingface.co/facebook/nllb-200-distilled-1.3B
[12]with the exception of the ID-EN dataset as the target language and source-side matrix language are both Indonesian. The embedded language English was treated as the source language instead.

1. Replacing the Fast Align tool (Dyer et al., 2013) with GIZA++ (Och and Ney, 2003) and custom bilingual dictionaries during the word alignment phase to improve overall alignment.

2. Expanding the analysis phase to include Named Entity Recognition (NER) and part-of-speech (POS) tagging for source sentences.

3. Adding new word substitution rules to the synthesis phase to take into consideration Named Entities and lexical properties based on POS from the previous step.

The expanded substitution process is summarised by Algorithm 1. In the pseudo-code, the bilingual word alignment result is represented by *bwa*, input sentence in the matrix language is *ms*, while its corresponding parsing tree, POS, and NER results are *pt*, *pos*, and *ner* respectively. The input sentence in the embedded language is *es*.

**Algorithm 1:** Synthetic code-switching data generation

**Data:** bwa, ms, es, pt, pos, ner
**Result:** Code-switched sentence
**begin**
    **for** *each Name Entity $e$ in ner* **do**
        **if** *translation of $e$ exists in es* **then**
            Replace $e$ in ms with its translation from es;
    **for** *each node $n$ in pt* **do**
        **if** *node $n$ is switchable according to Matrix Language Theory* **then**
            Set switch_label($n$) to True;
        **else**
            Set switch_label($n$) to False;
    **for** *each node $n$ with switch_label($n$) as True* **do**
        **if** *lexicality of $n$ is not in {noun, adjective, verb} based on pos* **then**
            Set switch_label($n$) to False;
    **for** *each node $n$ with switch_label($n$) as True* **do**
        **if** *translation of node's word exists in bwa and is in es* **then**
            Replace word of node $n$ in ms with its translation from es;
        **else**
            Continue without replacement;
    **return** *Modified ms as code-switched sentences*;

One of our main considerations while creating the synthetic dataset was to divest from the English-centric code-switch pairs in the real data. As such, we chose a translation direction containing no English in the code-switching source side and another with no English in both source and target. However, we were still constrained by the algorithm requiring word-level alignments between constituent languages, which made generating code-switch sentences with low-resource languages and with more than two languages difficult. We added the following four translation directions, including Tamil-English to Czech, where both Tamil and Czech may be considered low-resource. Each contains 1012 lines derived from the "devtest" split of Flores-200:

- English-Chinese (EN-ZH) to Chinese (ZH)
- German-Turkish (DE-TR) to English (EN)
- French-Italian (FR-IT) to Japanese (JA)
- Tamil-English (TA-EN) to Czech (CS)

| Language | Version | Sub Ratio (NOUN/ ADJ+VERB) | CMI |
|---|---|---|---|
| EN-ZH | V1 | 0.437 / 0.563 | 19.7 |
| | V2 | 0.333 / 0.667 | 18.8 |
| | V3 | 0.408 / 0.592 | 30.3 |
| DE-TR | V1 | 0.335 / 0.665 | 32.7 |
| | V2 | 0.234 / 0.766 | 33.0 |
| | V3 | 0.407 / 0.593 | 40.4 |
| FR-IT | V1 | 0.373 / 0.627 | 19.4 |
| TA-EN | V1 | 0.291 / 0.709 | 8.05 |

Table 1: Synthetic dataset properties by version.

By incorporating the additional POS data in the substitution, different versions of the EN-ZH and DE-TR data were generated. V1 is our standard dataset used for the overall benchmarking. Compared to V1, the POS distribution of the code-switching constituents for V2 is altered by reducing noun substitutions and increasing adjective and verb substitutions, while maintaining overall incidence of code-switching. In V3, the degree of code-switching is increased in comparison to V1 and V2. These differences are quantified with the code-mixing index (CMI) metric (Das and Gambäck, 2014) and the lexical substitution ratio between nouns and adjectives/verbs (Table 1). We utilised the different versions for further experiments with GPT-4. All derived data and the companion code will be made available[1].

| | Prompts |
|---|---|
| P1 | `Translate the following code-switched [SRC] sentences to pure [TGT] line by line. Do not output any additional text other than the translations: \n [SRC1] \n [SRC2] ...` |
| P2 | `Translate the following [SRC] sentences to pure [TGT] line by line. Do not output any additional text other than the translations: \n [SRC1] \n [SRC2] ...` |
| P3 | `Please provide the [TGT] translation for these sentences line by line. Do not output any additional text other than the translations: \n [SRC1] \n [SRC2] ...` |

Table 2: Candidate prompts. `\n` denotes a newline while `[SRC1]` and `[SRC2]` are source sentences.

### 2.4. Evaluation metrics

We use BLEU (Papineni et al., 2002) as the primary metric for translation quality, and compliment it with ChrF++ (Popović, 2017) and TER (Snover et al., 2006) which may be more representative for character-based languages like Chinese. Higher BLEU and ChrF++ scores are indicative of better translations while lower TER scores show the same. All metrics were calculated using SacreBLEU (Post, 2018) with lowercase settings, SacreBLEU's language-specific tokenizers and *"ter-asian-support"* flag for Japanese and Chinese.

### 2.5. Prompting strategy

To narrow down several candidate prompts for code-switching translation, we modified the initial prompt used by Jiao et al. (2023) for monolingual translation with the following: "Provide ten concise prompts or templates that can make you translate code-switched sentences.". We then identified the similarities between the candidates provided by GPT-3.5 and GPT-4, namely certain important keywords like the task "translate" together with the auxiliary "code-switched", and the source and target languages denoted by [SRC] and [TGT] respectively. The [SRC] is a composite of the languages in the code-switch with the matrix language coming first, for example "Spanish-English". This process guided us in narrowing the candidate prompts to the three in Table 7 by discarding those that were similar. In addition, we found that certain LLMs tended to append extra commentary to the translation so prompts were extended with instructions to mitigate this behaviour.

Notably, there were failure cases where the LLM would not output in the target language, merely restate the prompt, or state that it is unable to carry out the task. This behaviour was usually not consistent across repeated attempts, a likely effect of the stochastic sampling during generation. To handle such cases we retry the prompt up to a maximum of four attempts. Outputs that were still considered irregular after the retries were replaced with "-" before calculation of the translation metrics. Therefore, a low BLEU score in our overall benchmarking may be indicative of not only poor translation ability but also a failure to carry out the given task.

| Prompts | BLEU | ChrF++ | TER |
|---|---|---|---|
| P1 | **37.70** | 56.18 | 52.53 |
| P2 | 37.50 | **56.22** | **51.86** |
| P3 | 36.98 | 55.61 | 54.98 |

Table 3: Aggregated scores over three datasets and six models for the three candidate prompts considered.

Using a subset of 100 random lines each from the three real datasets, we evaluated the translation performance with the candidate prompts. Table 3 shows the results averaged across all six models. Results were fairly close between the prompts, with P2 slightly edging out the others on two out of three metrics. We subsequently adopt P2 for all other experiments, with slight variations when using the more advanced prompting techniques introduced later on[13]. The benchmarking was carried out in a zero-shot manner.

## 3. Results and Discussion

### 3.1. Overall benchmarking

**Relative LLM performance** The overall performance of various LLMs across the real and synthetic datasets is summarised in Table 4. Among the LLMs, GPT-4 clearly outperforms the others, followed by GPT-3.5 which lags behind GPT-4 by a minimum of 1.5 BLEU on FR-IT→JA to a maximum of 12.3 BLEU on TA-EN→CS. We found GPT-4's overall translation better than GPT-3.5 in terms of accuracy. However, GPT-3.5 may at times generate more natural translations in terms of sentence structure. Bard's performance showed significant variation across different datasets. It appeared to be comparable to GPT-3.5 on SP-EN, EN-ZH, DE-TR, and FR-IT translations, while outperforming on TA-EN but performing worse on HI-EN and ID-EN transla-

[13] Additional prompts are shown in Appendix A

| Model | HI-EN→EN | | SP-EN→EN | | ID-EN→ID | | *EN-ZH→ZH* | | *DE-TR→EN* | | *FR-IT→JA* | | *TA-EN→CS* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BLEU | ChrF++ | BLEU | ChrF++ | BLEU | ChrF++ | BLEU | ChrF++ | BLEU | ChrF++ | BLEU | ChrF++ | BLEU | ChrF++ |
| GPT-4 | **37.8** | **60.4** | **53.9** | **71.2** | **57.3** | **74.1** | 44.9 | 30.2 | **45.4** | **67.6** | 25.9 | 26.1 | 19.1 | 45.2 |
| GPT-3.5 | 30.5 | 54.9 | 48.6 | 69.1 | 48.7 | 66.6 | 41.1 | 27.1 | 43.1 | 66.5 | 24.4 | 25.0 | 6.8 | 29.5 |
| Bard-PaLM2 | 23.9 | 42.1 | 45.6 | 61.6 | 28.9 | 49.8 | 44.0 | 30.3 | 43.1 | 66.6 | 24.9 | 24.6 | 15.8 | 38.5 |
| LLaMA-2-70B | 25.7 | 49.4 | 40.2 | 61.2 | 34.3 | 50.4 | 34.4 | 23.6 | 37.2 | 60.8 | 19.9 | 20.5 | 0.9 | 16.6 |
| Falcon-40B | 5.9 | 25.0 | 15.4 | 34.4 | N/A | N/A | 20.8 | 15.3 | 25.6 | 52.1 | 1.3 | 4.7 | N/A | N/A |
| Phoenix-7B | 7.0 | 30.2 | 31.9 | 51.9 | 28.2 | 40.2 | 39.4 | 27.3 | 17.8 | 40.6 | 6.1 | 9.2 | 1.5 | 16.2 |
| Google T | 28.5 | 51.6 | 49.1 | 69.4 | 54.6 | 70.0 | **47.5** | **35.0** | 27.7 | 50.3 | **26.5** | 25.5 | **22.4** | **48.0** |
| DeepL T | N/A | N/A | 47.6 | 68.1 | 52.7 | 69.1 | 46.4 | 34.6 | 28.0 | 50.6 | 25.4 | **26.2** | N/A | N/A |
| NLLB-1.3B | 8.0 | 30.6 | 46.7 | 67.0 | 53.5 | 69.4 | 28.2 | 19.7 | 32.8 | 55.6 | 15.8 | 19.6 | 15.6 | 40.5 |
| NLLB-54B | 10.4 | 29.9 | 47.1 | 66.7 | 54.3 | 68.4 | 28.7 | 20.8 | 34.9 | 57.2 | 16.6 | 19.9 | 18.8 | 43.9 |
| Copy | 5.1 | 28.8 | 27.6 | 42.1 | 49.5 | 65.0 | 12.9 | 10.6 | 2.3 | 20.4 | 0.2 | 1.4 | 0.7 | 4.9 |

Table 4: BLEU and ChrF++ across various code-switching datasets for a collection of LLMs. They are evaluated against baselines containing commercial MT engines (Google and DeepL translate) and massive multilingual MT models (NLLB). "Copy" baseline are scores between untranslated source and reference target. Synthetic datasets are italicized.

tions. It also has a higher tendency for missing and mistranslations. Similarly, LLaMA-2's output contained frequent incidences of untranslated words from the source sentence. Comparatively, Falcon and Phoenix significantly underperformed the others. Falcon had a particularly high failure rate (see section 2.5) for several language directions but especially for ID-EN and TA-EN, for which we consider it unable to handle. Phoenix's translation may sometimes sound unnatural as it may not have been exposed to much code-switching data. However, its emphasis on multilingual training data, particularly for Chinese, helped it outperform Falcon on several directions even with a much smaller parameter size.

Most LLMs struggled with the low resource language pair TA-EN→CS. We observed a tendency for models like Phoenix, GPT-3.5, and Llama-2 to translate TA-EN code-switch into English, a mix of English and Czech, or English followed by Czech, using the former as an intermediary pivot prior to the final translation. It is evident that despite exposure to diverse monolingual data during training, typical LLM training still lacks coverage of low-resource languages, resulting in weak comprehension.

**Comparative performance against baselines** In terms of the quantitative metrics, GPT-4 excelled in four out of seven datasets, particularly those involving high-resource languages (e.g. EN, SP, HI, DE) and when translating into English. Google Translate topped the remaining three datasets: EN-ZH, FR-IT, and TA-EN. Qualitatively, it matched GPT-4 for EN-ZH and FR-IT, and slightly surpassed it for TA-EN. Notably, despite lagging behind GPT-4 for high-resource languages, NLLB-54B showed comparable capabilities for the low-resource TA-EN→CS direction. We find commercial engines perform relatively well on examples with low code-mixing index (CMI), showing a smaller gap compared to LLMs for code-switching than for monolingual translation. We believe some naturally occurring examples of code-switching in the training data helps reinforce this ability in the supervised models. Nevertheless, as will be shown in the following section, performance greatly deteriorates as the degree of code-switching increases.

Overall, GPT-4 and GPT-3.5 exhibit robust code-switching translation abilities, comparable to or better than current commercial engines and large multilingual MT models like NLLB, particularly for high-resource language pairs. However, the performance gap narrows in low-resource settings, as language modelling proves less efficient than directly learning from paired data for translation. Our analysis underscores the significance of a language's coverage in the training data, as it directly correlates with the overall performance of an LLM in that language.

**Some caveats** Unlike for monolingual translation, a certain language may appear both in the source as part of the code-switch, and also in the target. Hence, it may be possible to achieve a relatively high score, as measured by BLEU/ChrF++, by merely copying the source. The scores between the untranslated code-switch source and the reference target is provided in Table 4 under the "Copy" row. SP-EN→EN and ID-EN→ID directions were particularly problematic with a high overlap between source and target. Notably, only GPT-4 achieved a better score than "Copy" for ID-EN→ID among the LLMs.

## 3.2. Effect of code-switching composition

To further understand the impact of the composition of code-switching entities within an utterance on translation, we compared results across variants of the synthetic datasets (see Table 1) together with monolingual translation baselines on GPT-4 and Google Translate, shown in Figure 1.

**Increasing degree of code-switching** Both GPT-4 and Google Translate display a similar deterioration in performance as code-switching increases from the monolingual baseline to V3, although the effect is much more extreme for the latter. We observe that Google Translate actually performs better than GPT-4 for all three monolingual baselines (EN→ZH, EN→DE, DE→EN), but BLEU significantly drops as code-switching is introduced. Given their speculated large training data, we expect both models to have been exposed to some instances of naturally occurring code-switching text, although V3's mixing proportion may have been higher than what the models have seen, resulting in reduced performance.

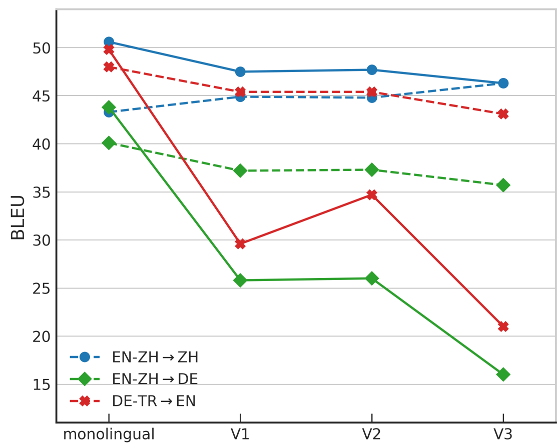


Figure 1: Trend in BLEU for GPT-4 (dashed line) and Google Translate (solid line) over different versions of the code-switching source, with fully monolingual versions on the far left.

As a traditional MT engine, Google Translate is limited to only supplying a single language as source, so mixing different languages may be perceived as added noise to the underlying model, resulting in sub-optimal utilisation of those parts of the input. While there was an attempt to translate the matrix language, a significant portion of the embedded sections were found to be untranslated. Conversely, the ability of LLMs (especially GPT-4) to understand they are being given code-switched inputs results in greater robustness towards higher occurrences of code-switching. Particularly, EN-ZH→ZH saw a gradual improvement for GPT-4. We attribute this to there being proportionally more of the target words in the source itself, allowing GPT-4 to use more of the reference vocabulary in its translations, thus achieving higher BLEU scores. This behaviour is different from Google Translate, which tended to restate the code-switched parts using different vocabulary.

**POS distribution of code-switching elements** Comparing V1 and V2 shows no significant impact of the POS distribution on translation performance across all three directions. This is also reflected in our qualitative evaluation, where the output retains similar translation quality and naturalness, and only displays slight differences in word choice and sentence structure, particularly for GPT-4. The exception was DE-TR→EN for Google Translate, for which we found a higher occurrence of untranslated words in V1 than V2 that may explain the gap in BLEU. Maintaining translation ability regardless of the actual distribution of language mixing is highly indicative of inherent cross-lingual ability and shows LLMs have the potential to improve even further with more explicit training procedures or data in this regard.

## 3.3. Improving code-switching translation ability

To better leverage the power of LLMs requires careful engineering of the input prompts. From the above investigation, it is evident that the performance of GPT-4 far surpasses that of other LLMs. Below we investigate more advanced prompting techniques with GPT-4 to explore the upper bounds on the translation capabilities of LLMs. Note that the following experiments were conducted on a random subset of 100 lines for each dataset, and so results might differ slightly to the overall benchmarking.

### 3.3.1. In-context learning

In-context learning augments the prompts to include demonstrations of the task of interest. It has been shown to boost performance over zero-shot prompting in lieu of finetuning the model. Previous work have demonstrated the benefits of in-context learning for monolingual translation (Agrawal et al., 2023; Hendy et al., 2023; Zhu et al., 2023), and we demonstrate below that it is similarly advantageous for code-switching inputs. We approach this experiment with a view that the test set is not known beforehand, a situation common in MT development, so selecting examples that are semantically similar to members of the test set like was carried out in previous work was not pursued. Instead, we investigated different strategies for in-context example selection that only considers the properties of the test set as a whole, without requiring information on specific sentences.

Potential examples were selected from the disjoint of the subset of test data employed, referred

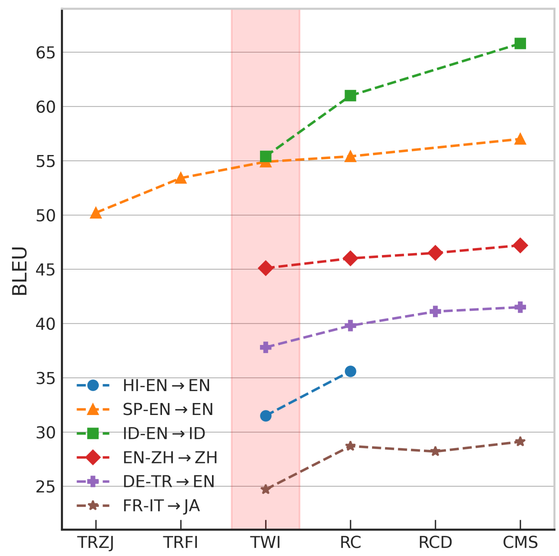


Figure 2: Trend in BLEU across different in-context learning methodologies. The translation baseline without in-context learning (TWI) is highlighted in red.

to as the candidate set. Following Zhang et al. (2023a), 10 samples were chosen following each selection strategy and appended to prompt P2. We compare the efficacy of each method in Figure 2. TWI, highlighted in red, is our baseline of zero-shot translation, that is without in-context learning.

**Task-related examples** Task-related examples contain code-switched sentences and their translation but with languages distinct from those in the test set. This is similar to the cross-lingual exemplars used by Zhu et al. (2023) who found to negatively impact monolingual DE to EN performance but enhance low-resource ZH to EN. We similarly found a detrimental effect on SP-EN→EN relative to the baseline. Moreover, the degree of linguistic divergence from the test language may affect results. For instance, samples from FR-IT→EN (TRFI), also Romance languages like Spanish, resulted in a BLEU score 1.5 points lower than the baseline compared to the more distant ZH-JA→EN (TRZJ) that lowered BLEU by 4.9, as depicted in Figure 2.

**In-domain examples** In-domain examples are sourced from the same type of data as the test set, thus sharing both the task and the translation direction. From Figure 2, these examples were beneficial with even randomly chosen ones (RC) resulting in significant improvements over the baseline, averaging a 2.85 BLEU increase, consistent with prior monolingual translation findings.

Given the challenges in acquiring code-switching sentence pairs for examples, we explored using monolingual translations from the matrix language of the code-switch instead (RCD). Here, RC samples were converted to their corresponding monolingual counterparts on the source side while keeping the target translations intact. This was done only on the synthetic datasets where such data was available. Based on the RCD results, providing in-context sentence pairs from the dominant language to the target language yields comparable or superior translations to using the code-switching examples directly. This may be attributed to monolingual translation being more familiar to the LLM, thereby enhancing its understanding of the task.

**Criterion-based examples** Inspired by the criteria for data selection in domain adaptation for machine translation, we introduce a novel example selection strategy based on diversity and exemplarity. To enhance sample diversity, we initially use the multilingual RoBERTa model (Liu et al., 2019) to embed source sentences from the candidate set. HI-EN was omitted due to RoBERTa only recognising Devanagari instead of the Latin script for Hindi. Employing the Affinity Propagation clustering algorithm (Frey and Dueck, 2007), we cluster sentence embeddings into approximately 10 classes, ensuring intra-class similarity and inter-class diversity.

Drawing on evaluative metrics employed to characterise code-switching data like CMI and others in Srivastava and Singh (2021), we contend that source sentences featuring a higher number of "switch points", defined as a token in the text that is preceded by a token in a different language, serve as more informative exemplars for the model. Utilising this insight, we select samples with the maximum switch points from each of the preceding clusters to be used as in-context learning examples. We term this method Clustering-Max Switching (CMS). Results (see Figure 2) demonstrate its effectiveness in choosing examples for code-switching translation, generally outperforming other strategies.

**Ablation study of CMS** An ablation study was carried out to investigate the relative importance of the two main steps in the CMS strategy. The results in Table 5 highlight that selecting sentences based on the maximum switch points across the entire candidate set (MS) without pre-clustering broadly improves BLEU compared to a purely random selection (RC) – from a modest 0.1 to a substantial 3.1 across all five language pairs. Meanwhile, sampling from clustered sentence embeddings randomly (CL) instead of choosing ones with the maximum switch points boosts BLEU scores

| Language | RC | MS | CL | CMS |
|---|---|---|---|---|
| SP-EN | 55.4 | 56.7 (+1.3) | 55.6 (+0.2) | **57 (+1.6)** |
| ID-EN | 61.0 | 64.1 (+3.1) | 61.9 (+0.9) | **65.8 (+4.8)** |
| ZH-EN | 46.0 | 47.0 (+1.0) | 46.1 (+0.1) | **47.2 (+1.2)** |
| DE-TR | 39.8 | 40.3 (+0.5) | 40.2 (+0.4) | **41.5 (+1.7)** |
| FR-IT | 28.7 | 28.8 (+0.1) | 28.0 (-0.7) | **29.1 (+0.4)** |

Table 5: Ablation study of CMS utilising only maximum switch points (MS), only clustering (CL) and the full method combining both (CMS). BLEU is reported relative to a baseline of randomly chosen examples (RC).

by 0.1 to 0.9 relative to RC for four language pairs, excepting FR-IT. When merging the two methodologies (CMS), the combined effect leads to an uplift of 0.4 to 4.8 BLEU across the five language pairs.

### 3.3.2. Pivot translation

The pivot strategy breaks the translation task into two steps: initially re-writing the source utterance in the pivot language, then translating it into the target language. For monolingual translation, pivoting may improve performance between languages with limited parallel data by linking them through a third high-resource language (Kim et al., 2019). Research has demonstrated significant improvement in LLM-based translation results by pivoting to English (Jiao et al., 2023; Zhang et al., 2023a). For LLMs, merging the pivot and final translations using a single prompt allows for extra context before the final translation. Apart from English, we adapt the pivot strategy for code-switching by investigating pivoting to the matrix language, essentially converting the code-switching input to its monolingual counterpart.

Comparing direct and pivot translation results (Table 6) confirms the effectiveness of pivoting, aligning with prior research. Generally, pivoting via English proves more effective than using the matrix language, as observed in both FR-IT→JA and TA-EN→CS cases, likely due to English's prevalence in LLM training data. Pivoting to the matrix language can still be effective if it is high-resource, as seen in DE, FR, and EN cases, but may instead deteriorate results for low-resource languages like TA. Double pivoting, i.e. via the matrix language first and then English, yields intermediate results. The pivoting technique is particularly beneficial for low-resource languages like TA and distant translation pairs like FR-IT to JA, where parallel data is limited. However, when the target is already high-resource like English, pivoting to the matrix language first may not be as effective, exemplified by the marginal improvement in BLEU observed in the DE-TR pivot by 0.1.

| Direction | Pivot Matrix | Pivot EN | Result BLEU | Result ChrF++ | Result TER |
|---|---|---|---|---|---|
| DE-TR→EN | (direct) | | 45.1 | 67.6 | **36.5** |
| | ✓ | | **45.2** | **67.9** | 36.7 |
| FR-IT→JA | (direct) | | 25.1 | 27.7 | 62.9 |
| | ✓ | | 26.2 | 27.8 | 63.0 |
| | | ✓ | **28.5** | 26.2 | **60.2** |
| | ✓ | ✓ | 27.4 | **29.0** | 61.3 |
| TA-EN→CS | (direct) | | 16.3 | 41.0 | 69.5 |
| | ✓ | | 15.6 | 41.2 | 69.4 |
| | | ✓ | **17.5** | **43.2** | **66.4** |
| | ✓ | ✓ | 16.7 | 41.6 | 71.7 |
| EN-ZH→ZH | (direct) | | 44.4 | 28.7 | 42.3 |
| | ✓ | ✓ | **45.1** | **29.0** | **41.0** |

Table 6: Results for matrix and English language pivot translation strategies. Note that for EN-ZH→ZH the two strategies are equivalent.

## 4. Conclusion

This study offers a thorough evaluation of LLMs' performance in code-switching translation, assessing six models across seven datasets, including non-English-centric ones, for a comprehensive overview of their capabilities. GPT-4 exhibited superior performance across both high and low-resource language pairs, while other models showed varying ability depending on the translation direction. Commercial engines like Google and DeepL Translate performed well on select datasets, particularly when code-switching was minimal. GPT-3.5's performance closely followed GPT-4 in high-resource languages but was surpassed by supervised MT engines for low-resource language pairs. We demonstrated GPT-4's robustness in handling heavier code-switching text and variations in POS distribution of code-switching elements. Additionally, we showed that translation capabilities could be enhanced through careful prompt engineering utilising in-context learning, in particular with our proposed CMS selection strategy, and pivot translation, especially to English. We anticipate this study will encourage greater efforts to incorporate cross-lingual abilities in LLMs, given their considerable potential for growth in this domain.

## 5. Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, and the Ministry of Communications and Information, under its Online Trust and Safety (OTS) Research Programme (MCI-OTS-001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Prime Minister's Office, Singapore, or the Ministry of Communications and Information.


## 6. Bibliographical References


Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: An open large language model with state-of-the-art performance.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*.

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

Myers-Scotton Carol. 1993. Duelling languages: Grammatical structure in codeswitching. *Claredon, Oxford*.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Phoenix: Democratizing ChatGPT across languages. *arXiv preprint arXiv:2304.10453*.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the*

*2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Yuan Gao, Ruili Wang, and Feng Hou. 2023. How to design translation prompts for ChatGPT: An empirical study. *arXiv preprint arXiv:2304.02182*.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? A comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT a good translator? Yes with GPT-4 as the engine. *arXiv preprint arXiv:2301.08745*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-English languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 866–876, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Carol Myers-Scotton. 1993. Dueling languages: Grammatical structure in code-switching.

NLLB-Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Shana Poplack. 1978. *Syntactic structure and social function of code-switching*, volume 2. Centro de Estudios Puertorriqueños, City University of New York.

Maja Popović. 2017. ChrF++: Words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Vivek Srivastava and Mayank Singh. 2021. Challenges and limitations with the metrics measuring the complexity of code-mixed text. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Orion Weller, Matthias Sperber, Telmo Pires, Hendra Setiawan, Christian Gollan, Dominic Telaar, and Matthias Paulik. 2022. End-to-end speech translation for code switched speech. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1435–1448, Dublin, Ireland. Association for Computational Linguistics.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.

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 Aji. 2023. Prompting multilingual large language models to generate code-mixed texts: The case of south East Asian languages. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63, Singapore. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Aji. 2023b. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

## 7. Language Resource References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. 2019. Normalization of Indonesian-English code-mixed Twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, Hong Kong, China. Association for Computational Linguistics.

Deuchar, Margaret and Davies, Peredur and Herring, Jon Russell and Parafita Couto, M and Carter, Diana. 2014. *Building Bilingual Corpora*. Multilingual Matters, Advances in the Study of Bilingualism.

# A. Prompt templates

| Experiment | **Prompts** |
|---|---|
| In-context learning (sec 3.3.1) | `Translate the following [SRC] sentences to pure [TGT] line by line. Here are some translation examples for your reference. \n [SRC]: [sample_source_sentence1]; [TGT]:[sample_target_sentence1] \n ... Do not output any additional text other than the translations: \n [SRC1] \n [SRC2] \n ...` |
| Pivot translation (sec 3.3.2) | `Translate the following [SRC] sentences to pure [SRC_matrix] first and then to [TGT] line by line. Do not output any additional text other than the translations including bullet points. \n [SRC1] \n [SRC2] \n ...` |
| | `Translate the following [SRC] sentences to pure English first and then to [TGT] line by line. Do not output any additional text other than the translations including bullet points. \n [SRC1] \n [SRC2] \n ...` |
| | `Translate the following [SRC] sentences to pure [SRC_matrix] first then to English, and finally to [TGT] line by line. Do not output any additional text other than the translations including bullet points. \n [SRC1] \n [SRC2] \n ...` |

Table 7: Modified prompt templates used in section 3.3. `\n` denotes a newline, `[SRC]` and `[TGT]` are source (*matrix-embedded*) and target language respectively, `[SRC_matrix]` is the matrix language of the source codeswitch, and `[SRC1]` and `[SRC2]` are source sentences.