Replicable Benchmarking of Neural Machine Translation (NMT) on Low-Resource Local Languages in Indonesia

Lucky Susanto[§,♠]
lucky.susanto@ui.ac.id

Ryandito Diandaru[§,◇]
13519157@std.stei.itb.ac.id

Adila Krisnadhi[♠]
adila@cs.ui.ac.id

Ayu Purwarianti[◇]
ayu@itb.ac.id

Derry Wijaya[♣,♡]
derry.wijaya@monash.edu

[♠]University of Indonesia [◇]Bandung Institute of Technology [♣]Boston University

[♡]Monash University Indonesia

Abstract

Neural machine translation (NMT) for low-resource local languages in Indonesia faces significant challenges, including the need for a representative benchmark and limited data availability. This work addresses these challenges by comprehensively analyzing training NMT systems for four low-resource local languages in Indonesia: Javanese, Sundanese, Minangkabau, and Balinese. Our study encompasses various training approaches, paradigms, data sizes, and a preliminary study into using large language models for synthetic low-resource languages parallel data generation. We reveal specific trends and insights into practical strategies for low-resource language translation. Our research demonstrates that despite limited computational resources and textual data, several of our NMT systems achieve competitive performances, rivaling the translation quality of zero-shot gpt-3.5-turbo. These findings significantly advance NMT for low-resource languages, offering valuable guidance for researchers in similar contexts.

1 Introduction

Neural Machine Translation (NMT) holds a crucial role for local languages in Indonesia, supporting language documentation (Abney and Bird, 2010), native language preservation (Bird and Chiang, 2012; Costa-jussà et al., 2022), and bridging socioeconomic gaps (Azzizah, 2015). However, challenges unique to low-resource languages have hindered progress in this field (Aji et al., 2022). Our work addresses these challenges for four prominent local languages in Indonesia: Javanese, Sundanese, Minangkabau, and Balinese.

Impressive NMT advancements often come from well-resourced entities (i.e., Google's PaLM2 (Anil et al., 2023), OpenAI's gpt-3.5-turbo (Brown et al., 2020), Facebook's NLLB-200 (Costa-jussà et al., 2022)), focusing primarily on high-resource languages like English. This phenomenon highlights a research gap for languages with limited resources in data availability and computing power. For instance, benchmark NMT systems like NLLB-200 (Costa-jussà et al., 2022) rely on substantial computing power, a luxury many researchers lack, especially those working with local Indonesian languages (Cahyawijaya et al., 2022). This hampers progress due to the difficulty of gauging whether a new approach, method, architecture, or data augmentation would help improve model performance.

In this work, our contribution is a replicable benchmark of NMT systems for these local Indonesian languages trained on publicly available data and tested on the publicly available FLORES-200 dataset. We prioritize accessible computing resources. Our base cross-lingual (Conneau and Lample, 2019) XLM model uses only a modest compute setup. It is trained with only two languages at a time and on a single GPU with at most 48GB of memory, which we believe is within the reach of most researchers in this domain. We also only use publicly available data sources to train, including the NusaCrowd repository of Indonesian languages (Cahyawijaya et al., 2022) and parsed wikidumps results [1]. Extending prior work, such as (Winata et al., 2023), we benchmark NMT models on multiple domains.

In addition, in a preliminary study, we explore the impact of using gpt-3.5 (Brown et al., 2020) for synthetic low-resource language data generation to augment training. We also investigate code-switching's potential (Kuwanto et al., 2021) for improving low-resource language NMT that was

[§]Equal contributions.

[1] Wikidumps Page

Proceedings of the First Workshop in South East Asian Language Processing, pages 100–115
November 1, 2023. ©2023 Association for Computational Linguistics

previously unexplored for Indonesian languages.

2 Related Work

2.1 NMT Benchmarks for Low-resource Local Languages in Indonesia

Neural Machine Translation (NMT) benchmarks are pivotal in documenting and preserving low-resource local languages like those in Indonesia. Prior works (Costa-jussà et al., 2022; Cahyawijaya et al., 2022; Winata et al., 2023) have contributed to the creation of these NMT Benchmarks in two significant ways: (1) the creation and compilation of datasets accompanied by (2) exploration and evaluation of different methodologies.

Costa-jussà et al. (2022) focused on developing NMT Benchmarks for low-resource languages. Their NLLB-200 model supports 200+ languages with more than 40K translation directions. Among these 200+ languages, some are local languages in Indonesia. They obtain state-of-the-art results for many translation directions through massive data collection efforts and computing resources.

Similarly, Winata et al. (2023) also collects a multilingual dataset for both machine translation and sentiment analysis for ten local languages in Indonesia. They use the collected dataset to create a benchmark for both tasks, obtaining impressive results in the machine translation tasks for the review domain by fine-tuning pre-trained models. Winata et al. (2023) also shows that fine-tuning a non-English-centric pre-trained model on local Indonesian languages outperforms its English-centric counterpart for the machine translation task.

However, it is essential to recognize that the NMT benchmark created by Costa-jussà et al. (2022) is challenging to replicate. Many researchers and institutions, including top Indonesian universities (Cahyawijaya et al., 2022), do not have access to massive compute resources or extensive and proprietary training data required to train the NLLB-200 models. Meanwhile, the NMT benchmark created by Winata et al. (2023) is limited only to the review domain. These leave a research gap that needs to be filled by benchmark NMT models that are replicable and cover more general domains.

2.2 High-resource vs Low-resource NMT

Unlike low-resource NMT systems (where either the source or target language is a low-resource language), NMT systems for high-resource languages have achieved impressive results (Costa-jussà et al., 2022). Even with the progress achieved by the grassroots movement mentioned in the previous section, the performance gaps are wide. This phenomenon is due to research in the field of NMT and NLP being dominated by English and other major languages, which means that more efforts have significantly been put into developing language technologies for these significant languages, and more data have been collected and made available for these languages (Akhbardeh et al., 2021; Kocmi et al., 2022). This means that while low-resource NMTs face problems no longer found in high-resource NMTs, insufficient resources and attention are being allocated.

One significant issue NMT systems face is the pivotal role parallel data plays in model performance (Koehn and Knowles, 2017). By definition, low-resource languages have little to no parallel data. One problem that negatively impacts the model performance is out-of-vocabulary (OOV) occurrences (Aji et al., 2022; Wibowo et al., 2021), where the model needs to see a token more to learn what it means. While this issue exists even in high-resource languages, the rate of occurrence for low-resource languages is substantially higher. However, usage of byte pair encoding (BPE) (Sennrich et al., 2016b) is capable of alleviating this issue to some degree (Lample et al., 2018b; Yang et al., 2020).

While previous research has made noteworthy strides in addressing out-of-vocabulary (OOV) occurrences, the most effective solution continues to be expanding available training data. However, building textual resources for translation tasks necessitates a significant investment of money, time, and expertise. Because of this, current research increasingly centers on finding innovative ways to augment the training of NMT models.

2.3 Augmenting Training for NMT

To combat the issue of data starvation, many researchers aim to utilize monolingual data to train NMT systems (Lample et al., 2018a; Artetxe et al., 2018; Conneau and Lample, 2019) and find ways to generate more training data, either comparable or synthetic data. Comparable data are extracted using various bitext retrieval methods (Zhao and Vogel, 2002; Fan et al., 2021; Jones and Wijaya, 2021; Kocyigit et al., 2022), multimodal signals (Hewitt et al., 2018; Rasooli et al., 2021), dictionary- or knowledge-based approaches (Wijaya and Mitchell, 2016; Wijaya et al., 2017; Tang and Wijaya, 2022); while synthetic data are created and utilized either through innovative training data augmentation (Kuwanto et al., 2021), utilizing automatic back-translation (Sennrich et al., 2016a; Wang et al., 2019), or even outright generating synthetic data using generative models (Lu et al., 2023), which has gained increasing attention by the community lately due to the advancement of large language models (LLMs).

Both Artetxe et al. (2018) and Lample et al. (2018a) show that NMT systems can be trained using only monolingual data while achieving impressive results. Conneau and Lample (2019) then create the XLM architecture, which allows NMT systems to be traine using monolingual and parallel data. Afterward, (Kuwanto et al., 2021) exploits the cross-lingual nature of the XLM models by corrupting the monolingual data using code-switching, which makes a single training instance contain multiple languages. The result is an improvement in the model's performance for low-resource translation.

In addition, prior works also focus on obtaining synthetic training data by turning monolingual data into parallel data through automatic back-translation (Sennrich et al., 2016a) or by using LLMs such as the gpt-family models (Brown et al., 2020) that have been gaining popularity in recent years in many fields (Lu et al., 2023). While back-translation has evolved, becoming a prominent method in the field of NMT (Artetxe et al., 2018; Conneau and Lample, 2019), using LLMs to generate synthetic data has yet to be thoroughly explored. This trend of using generative AI to generate synthetic training data displays initial potential, considering their remarkable performances compared to the state-of-the-art in machine translation (Zhu et al., 2023). However, further research with ablation studies and the inclusion of more language coverage is still needed.

3 Methodology

In this section, we outline our methodology for creating a replicable NMT benchmark for four Indonesian languages: Javanese (jv), Sundanese (su), Minangkabau (min), and Balinese (ban). We aim to systematically explore different training approaches and paradigms for NMT while maintaining a consistent base architecture (XLM), fixed hyperparameters, and controlled computing environment. Our compute environment is given a strict upper bound, in which a total of 48 GPU Hours from a single GPU for each model training, totaling up to 96 GPU Hours for NMT systems utilizing pre-trained language models. We also limit the memory of the GPU used to a maximum of 48 GB.

3.1 Training Approaches

We employ three primary training approaches to build our NMT models:

From Scratch (Scratch): In this approach, models are trained from the ground up without any reliance on pre-existing pre-trained language models. This approach acts as a baseline and allows us to gauge the performance of the models when trained from scratch.

Pre-trained Cross-Lingual (PreXL): Here, an NMT model utilizes a pre-trained cross-lingual model (XLM) (Conneau and Lample, 2019) on two sets of monolingual data. One of the sets is the Indonesian monolingual data, and the other is the low-resource local language monolingual data. This provides a strong starting point for the NMT by initializing the model with knowledge from the target and source languages. Therefore, each language pair in this work is given its own respective model. The number of pre-trained models for **PreXL** equals the number of language pairs in our work, which is four.

Code-switched Pre-trained Cross-Lingual (CodeXL): This approach involves pre-training the language model using additional augmented data from the two sets of monolingual data and a bilingual dictionary through code-switching, explained later in section 3.6. Code-switching allows for a bilingual context within each training instance. The pre-trained model is then fine-tuned for translation. The tasks used to fine-tune **CodeXL** and **PreXL** depend on the training paradigm used (section 3.2. The number of pre-trained models for **CodeXL** is the same as **PreXL**, which is four.

We chose the XLM architecture due to its modest compute resource requirements and its capability of cross-lingual language modeling. Moreover, the architecture is widely used for many low-resource language pairs and shows impressive results despite its modest size (Wang et al., 2019). We use Masked Language Modeling (MLM) (Devlin et al., 2019) to pre-train all the XLM models.

3.2 Training Paradigms

Additionally, we also explore two training paradigms, each influencing how the NMT models learn and what data are used for training:

Unsupervised NMT (Unsup): This paradigm trains the NMT system using only monolingual data of the source and target language. Utilizing both denoising-autoencoding (Vincent et al., 2010) and automatic back-translation (Sennrich et al., 2016a) to train the NMT system. Note that even though **CodeXL** utilizes a bilingual dictionary, it does not use any parallel data during pre-training.

Semi-supervised NMT (Semisup): This paradigm trains the NMT system using both monolingual data and parallel data of the source and target language. Monolingual data are utilized for training NMT by automatic back-translation.

We employ these shortened terms throughout our experiments to refer to the respective training approaches and paradigms. Our results indicate that the performance of each combination of approaches and paradigms on the evaluation dataset depends heavily on the amount of available data for the language: **Unsup** paradigm works better for very low-resource languages. In contrast, **Semisup** paradigm performs better when at least 10K parallel data is available (Artetxe et al., 2018). We do not conduct training using a strictly **Supervised NMT** paradigm because prior work has shown automatic back-translation's undeniable impact in improving low-resource NMT systems performance (Sennrich et al., 2016a).

3.3 Training with Synthetic Data

Following recent trends of using generative AI to generate synthetic training data (Lu et al., 2023; Zhu et al., 2023), we explore the impact of synthetically generated data on low-resource language NMT systems. We define two main approaches to generating synthetic data: (1) generating parallel data using generative AI and (2) translating monolingual data using an existing model.

To gauge the impact of the synthetically generated training data, we train NMT systems with these additional data using the **Scratch** and **CodeXL** training approaches. **Scratch** is also used in our preliminary experiments to identify the synthetic data generation approach that would yield the best empirical results. Once we identify the best approach, we apply the same synthetic data generation approach to all our language pairs and use the generated data to augment the training of our NMT approach with the **Semisup** paradigm.

Through the preliminary experiments (reported in Appendix C), we find that synthetic data generated using generative AI (gpt-3.5-turbo) has the most positive impact on training NMT systems. We generate 5000 parallel sentences for each language pair via a zero-shot prompt[2]: ***"Generate a long parallel sentence in SRC and TGT"***, where **SRC** and **TGT** is the pair of language we want to generate the sentences in. Appendix C provides justifications for these choices.

3.4 Fine-tuning Objectives

Denoising autoencoding **(DAE)** (Vincent et al., 2010) is a popular training objective for fine-tuning pre-trained LM for unsupervised MT tasks (Lample et al., 2018b; Wang et al., 2019) for its ability to increase the robustness of NMT models.

By utilizing the XLM architecture, our NMT system can perform multi-way translation. Thus, we also utilize automatic back translation **(BT)** (Sennrich et al., 2016a) during fine-tuning of our NMT models with **Unsup** and **Semisup** paradigms. By performing back translation using the same model that is being trained, synthetic parallel data is obtained and used automatically during training.

3.5 Training Data

Lang	Mono	Para
jv	1.6M	14.3K
su	550K	13.2K
min	282K	17.2K
ban	60K	0.9K

Table 1: Total number of monolingual sentences (**Mono**) and parallel sentences paired with Indonesian (**Para**) per language

We obtain our monolingual data from multiple publicly available sources. For Indonesian (id), we use the 201M monolingual sentences available from the Indo4B curated dataset (Wilie et al., 2020). We obtain monolingual data for the local languages through publicly available data such as Wikidumps[3], cc100 (Conneau et al., 2020), imdb-jv (Wongso et al., 2021), jadi-ide (Hidayatullah et al., 2020), and su-emot (Putra et al., 2020). The amount of monolingual sentences used to train each language is available in Table 1, with further breakdown available in Appendix A.

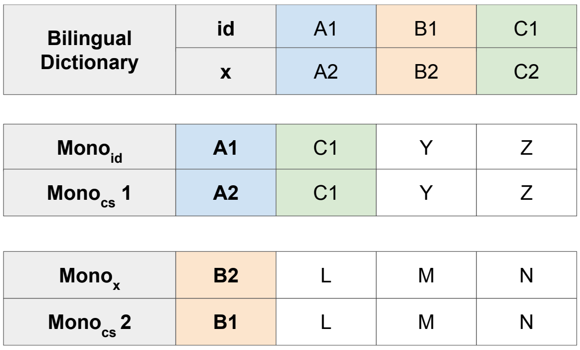
[2] Repository of the data we generate using zero-shot prompting

[3] Wikidumps Page

All parallel data we use to train the model are also publicly available from the NusaCrowd repository (Cahyawijaya et al., 2022). We scan the repository for datasets that contain parallel data of the local language paired with Indonesian. The amount of parallel sentences used to train each language pair is available in Table 1. Our largest language in terms of monolingual and parallel sentences, Javanese, is a tiny fraction (almost a 20th and a 500th, respectively) of NLLB-200 reported sentences for Javanese. From publicly available resources in Table 1, we can see that these four languages represent low-resource languages. A further breakdown is available in Appendix B.

The sentence counts in Table 1 are after we perform filtering on both monolingual and parallel data. For monolingual data, we remove sentences that contain less than three words or more than 250 words. We also perform simple filtering for sentences obtained from Wikipedia, including deduplication, removing HTML tags, removing sentences with only numbers, removing sentences that do not start with an alphabet, and removing metadata, bulletin points, or number ordering from sentences. For parallel data, we remove sentences that contain less than three words or more than 250 words and remove sentence pairs whose source sentences have a word count ratio above 1.5 of their translations following the setup of Ghazvininejad et al. (2023).

3.6 Code Switch



Bilingual Dictionary				
id	A1	B1	C1
x	A2	B2	C2

Mono_{id}	A1	C1	Y	Z
Mono_{cs} 1	A2	C1	Y	Z

Mono_{x}	B2	L	M	N
Mono_{cs} 2	B1	L	M	N

Figure 1: Illustration of generating synthetic data using code-switch. Using a bilingual dictionary, some words in each monolingual sentence are translated (i.e., blue and orange words). The candidate word for translation is chosen randomly, and not all words will be translated (green box).

In this paper, code-switching is done by utilizing the system made by Kuwanto et al. (2021)[4]. Code-switching is used to create synthetic training data by utilizing a bilingual dictionary. The generated data is used only during pre-training and is treated as a third language (labeled **cs**), where each training instance contains tokens from the other two languages. Results obtained by Kuwanto et al. (2021) imply that this method helps the model by giving stronger cross-lingual signals, which helps translation tasks during fine-tuning.

Lang	$\textbf{Mono}_{id}$	$\textbf{Mono}_{x}$	$\textbf{Mono}_{cs}$
{id, jv, cs}	201M	1.6M	209M
{id, su, cs}	201M	550K	208M
{id, min, cs}	201M	282K	208M
{id, ban, cs}	201M	60K	207M

Table 2: Total number of monolingual training instances for each language in each NMT system. $\textbf{Mono}_{id}$, $\textbf{Mono}_{x}$, $\textbf{Mono}_{cs}$ denotes the size of training instance for Indonesian, regional, and the third language, respectively.

Creating synthetic training data through code-switching utilizes training data from monolingual datasets from both languages in the system. By utilizing a bilingual dictionary, obtained and parsed from Winata et al. (2023)[5], each instance of training data from both monolingual datasets is augmented. Figure 1 illustrates this process, while Table 2 shows how much augmented training data is available for each NMT system.

4 Experiment Results

We concentrate our efforts on four language pairs: **id-jv**, **id-su**, **id-min**, and **id-ban**. Indonesia (**id**) is spoken by approximately 198 million people worldwide, whereas Javanese (**jv**), Sundanese (**su**), Minangkabau (**min**), and Balinese (**ban**) are spoken by roughly 68.2 million, 32.4 million, 4.8 million, and 3.3 million people, respectively, according to (Eberhard et al., 2023). Unsurprisingly, monolingual and parallel data availability for these four local languages in Indonesia generally follows a similar pattern. Javanese boasts the most extensive corpus of monolingual text data, while Balinese has the smallest. Regarding parallel data, Sundanese leads the way, closely followed by Javanese, while Balinese trails behind. Due to the substantial variation in training data availability, we present our findings for each local language separately. This approach allows us to assess the impact of different

[4] Code Switch-based Curriculum Training

[5] NusaX's bilingual dictionary

Translation	Paradigm	Turbo	Scratch	Scratch$_{AUG}$	PreXL	CodeXL	CodeXL$_{AUG}$
id-->jv	Zero shot	18.91	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	12.06	–.–	16.94	*17.90*	–.–
	Semisup	–.–	13.50	18.29	19.31	***21.32***	21.18
jv-->id	Zero shot	**29.97**	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	08.11	–.–	13.73	*20.93*	–.–
	Semisup	–.–	12.18	21.36	18.88	26.17	*26.23*
id-->su	Zero shot	16.39	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	10.42	–.–	*14.69*	10.68	–.–
	Semisup	–.–	13.19	15.33	16.22	***18.91***	18.90
su-->id	Zero shot	**30.71**	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	08.07	–.–	*13.62*	10.68	–.–
	Semisup	–.–	13.55	20.72	21.68	28.06	*28.40*
id-->min	Zero shot	13.71	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	10.18	–.–	16.03	*18.37*	–.–
	Semisup	–.–	22.33	22.59	23.83	***26.04***	25.18
min-->id	Zero shot	28.27	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	07.88	–.–	12.00	*19.43*	–.–
	Semisup	–.–	17.48	25.27	20.93	29.83	***30.06***
id-->ban	Zero shot	**14.94**	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	08.28	–.–	11.03	*12.70*	–.–
	Semisup	–.–	00.22	00.30	02.63	06.36	*09.51*
ban-->id	Zero shot	**26.93**	–.–	–.–	–.–	–.–	–.–
	Unsup	–.–	07.13	–.–	10.34	*18.05*	–.–
	Semisup	–.–	00.30	00.36	05.35	11.16	*17.34*

Table 3: Performance of NMT systems for Indonesian (**id**), Javanese (**jv**), Sundanese (**su**), Minangkabau (**min**), and Balinese (**ban**) translations. **Turbo** refers to gpt-3.5-turbo's zero-shot translation performance. **Zero shot** paradigm indicates translation without training. **AUG** denotes models trained with synthetic data generated by gpt-3.5-turbo. Bold values represent the best overall performance, while italicized values indicate the best performance within each paradigm.

training methods and paradigms while assessing the influence of training data size.

We conduct experiments using three training approaches (**Scratch**, **PreXL**, **CodeXL**) and two training paradigms (**Unsup**, **Semisup**). For each combination of these approaches and paradigms, four NMT systems are trained (one of each language pair mentioned above). In total, there are 24 different NMT systems trained this way.

In addition, we conduct experiments using synthetic parallel datasets. We only generate parallel training data, so these data do not affect the **Unsup** training paradigm. To evaluate the impact of these synthetic datasets, we employ two distinct training approaches: **Scratch** and **CodeXL**. We denote the process of training NMT systems with additional synthetic parallel training data as **Scratch**$_{AUG}$ and **CodeXL**$_{AUG}$, respectively. These comprise the remaining 8 NMT systems created in this work, totaling 32. Since we use gpt-3.5-turbo to generate our synthetic parallel data in a zero-shot manner, we also benchmark the zero-shot translation performance of gpt-3.5-turbo (**Turbo**) on our evaluation dataset.

The results of these experiments (all metrics are in spm200BLEU, shown in Table 3), reveal a consistent trend: **CodeXL** approach results in a significantly better performing NMT systems compared to **Scratch** and **PreXL**. An exception to this pattern is noted in the **id-su** language pair when employing the **Unsup** training paradigm.

4.1 Javanese

The **id-jv** language pair is particularly significant due to its relevance in Indonesia, where approximately 198 million people speak Indonesian (id), and Javanese (jv) is spoken by roughly 68.2 million (Eberhard et al., 2023).

Looking at Table 3, we observe a substantial gap in translation performance between **id**→**jv** and **jv**→**id**, emphasizing the performance asymmetry. Notably, when training NMT systems using the **Unsup** paradigm, **CodeXL** consistently outperforms other approaches for both translation directions, reinforcing the findings of Kuwanto et al. (2021), which highlight the generalization capability of this approach.

Surprisingly, when we train NMT models using additional parallel data generated by gpt-3.5-turbo (**CodeXL**$_{AUG}$), we notice a slight decline in performance for **id→jv** translation compared to our best-performing model (**CodeXL**). A more detailed comparison is discussed in the next section.

4.2 Sundanese

Table 3 shows where **id-su** differs from the **id-jv** language pair. For **id-su** language pair, the **Unsup** paradigm shows a different trend where **CodeXL** has a slightly worse performance compared to **PreXL** in the **id→su** translation. However, this trend shifts when using the **Semisup** paradigm, with **CodeXL** regaining its superiority.

Similar to **id-jv** language pair, an intriguing phenomenon arises when we train NMT models using additional parallel data generated by gpt-3.5-turbo (**CodeXL**$_{AUG}$) for the **id-su** language pair. While this approach does not create a better performing model in **id→su** translation, it does result in a slightly better model for **su→id**. This trend indicates that the generated synthetic parallel data's impact heavily depends on the generative AI's translation performance. For both **id-jv** and **id-su** language pairs, gpt-3.5-turbo's zero-shot translation performance on **id→x** is worse than **CodeXL** for each respective language pair, therefore **CodeXL**$_{AUG}$ does not result in improved performance. Meanwhile, the reverse is true, gpt-3.5-turbo's **x→id** translation performance is better than **CodeXL** in **x→id** direction, hence **CodeXL**$_{AUG}$ has a better performance in this direction.

4.3 Minangkabau

The results we obtained for **id-min** follow a similar pattern as **id-jv**. The trend where **CodeXL** models performed better than **Scratch** and **PreXL** continues for **id-min** for both translation directions.

However, unlike **id-jv** and **id-su**, using synthetically generated parallel data to train NMT systems for **id-min** (**CodeXL**$_{AUG}$) performed better than **CodeXL** on the **min→id** translation. This is surprising because **CodeXL** performed better than **Turbo** on **min→id**, yet the parallel data generated by **Turbo** was able to create **CodeXL**$_{AUG}$, which is a better performing NMT system. This breaks the previous trends set by **id-jv**, **id-su**, and even **id-ban** in the later section.

4.4 Balinese

id-ban continues the trend set by the majority of previous language pairs. Following **id-jv** and **id-min** language pairs, **CodeXL** consistently has superior performance compared to **Scratch** and **PreXL**.

Additionally, **id-ban** follows the trend set by **id-jv** and **id-su**, where the use of synthetically generated parallel data from **Turbo** creates a better NMT system compared to others that do not use them. For **id-ban** language pair specifically, **Turbo**'s translation performance is much higher than **CodeXL**, and the data **Turbo** generated has a significant impact during training, as seen in **CodeXL**$_{AUG}$. The difference in score for **CodeXL** and **CodeXL**$_{AUG}$ differs by **3+** and **6+** spm200BLEU for **id→ban** and **ban→id** respectively. This performance difference is much more significant compared to **id-jv**, **id-su**, and **id-min**, where the performance difference is less than **1** spm200BLEU. This finding supports the idea that the generated synthetic parallel data's impact heavily depends on the generative AI's translation performance. Moreover, if the initial parallel data is limited, like in the case of **id-ban** (only 0.9K), the addition of synthetic data can greatly improve performance.

However, Table 3 shows that **Unsup** training paradigm created the best performing NMT system for **id-ban** language pair. While it would not be surprising for **Scratch** due to the limited amount of parallel training data, it is surprising that **Scratch**$_{AUG}$ does not result in a considerably better NMT system, as the parallel training data size becomes 6x its original size (i.e., from 0.9K to 5.9K). This indicates that **denoising-autoencoding** plays a more significant role in model performance than parallel data when the training parallel data is limited.

5 Conclusion

In this work, we create a replicable NMT benchmark under low-resource settings. We comprehensively train and analyze NMT systems for four low-resource Indonesian local languages: Javanese, Sundanese, Minangkabau, and Balinese. Our experiments shed light on the impact of different training approaches, paradigms, data sizes, and generated synthetic parallel data in low-resource local languages in Indonesia. In conclusion:

We observe that the **CodeXL** training approaches generally create NMT systems with better performances compared to **Scratch** and **PreXL** approaches. This further strengthens the robustness of the approach suggested by Kuwanto et al. (2021), where code-switching is used to give a stronger cross-lingual signal during model pretraining. Code-switching more positively impacts translation performance for **x**→**id** more than **id**→**x**. For reference, NMT systems created using the **CodeXL** training approach and **Semisup** paradigm have an average performance of **23.80** and **18.15** spm200BLEU for **x**→**id** and **id**→**x** respectively.

Furthermore, even after pre-training a language model using the MLM objective, fine-tuning the model using the denoising autoencoding objective might play a more prominent role in extremely low-resource NMT than just training the model to be more robust for the translation task. This is shown in the **id-ban** language pair NMT systems, where **Unsup** created better NMT systems than the **Semisup** training paradigm. It is noteworthy that the addition of 5000 synthetic parallel training data might not be enough to significantly improve NMT system performance, as visible in the **CodeXL-Unsup** entry compared to **CodeXL**$_{AUG}$**-Semisup** entry in Table 3, since the resulting parallel data is still very limited (i.e., less than 6K sentences).

Lastly, we also observe a trend in which generative AIs can help augment the training process by generating synthetic parallel data. In most cases, excluding the **id-min** language pair, the parallel data generated by generative AI can impact the performance of NMT systems to approach or even outperform the performance of the generative AI's translation performance with much less compute and data resource.

6 Future Work

Along with the above conclusions, our work also opens several venues for future research. Further ablation studies are needed to fully understand the impact of denoising-autoencoding on translation tasks. Our results indicate that the denoising-autoencoding objective not only increases model robustness but may also play a role in cross-lingual language understanding in extremely low-resource NMT.

In addition, further investigations into synthetically generated parallel data quality and diversity are crucial. We observe a trend where synthetically generated parallel data from gpt-3.5-turbo impact the training of NMT systems such that its performance approaches or even outperforms gpt-3.5-turbo's zero-shot translation performance.

7 Limitation

While our work has given insight into NMT systems for low-resource local languages in Indonesia, it is essential to note that we have utilized different GPUs (TitanV, RTX8000, RTX6000, A6000, A40) with a maximum memory capacity of 48GB for different experiments. These GPU architectures and memory capacity variations may have influenced the observed performance. However, it is crucial to recognize that hardware differences alone cannot fully account for all the performance gaps observed. Future research should conduct experiments using a more standardized GPU setup to understand the impact of hardware variations better.

Additionally, all of our experiments that include the utilization of gpt-3.5-turbo are problematic as it is a closed-sourced model. This causes problems such as transparency and reproducibility in the future. Future work should continue performing ablation studies on open-sourced LLMs.

8 Acknowledgement

Authors from Indonesia are supported by the MoE-CRT ACE Open Research program. The authors also thank the Indonesian government for their funding and Boston University for providing essential computing resources. We also thanks Garry Kuwanto from Boston University for his help in utilizing his Code-switching system.

References

Steven Abney and Steven Bird. 2010. The human language project: Building a universal corpus of the world's languages. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 88–97.

Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

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.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yuni Azzizah. 2015. Socio-economic factors on indonesia education disparity. *International Education Studies*, 8:218.

Steven Bird and David Chiang. 2012. Machine translation for language preservation. In *Proceedings of COLING 2012: Posters*, pages 125–134.

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. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Samuel Cahyawijaya, Holy Lovenia, Alham Fikri Aji, Genta Indra Winata, Bryan Wilie, Rahmad Mahendra, Christian Wibisono, Ade Romadhony, Karissa Vincentio, Fajri Koto, Jennifer Santoso, David Moeljadi, Cahya Wirawan, Frederikus Hudi, Ivan Halim Parmonangan, Ika Alfina, Muhammad Satrio Wicaksono, Ilham Firdausi Putra, Samsul Rahmadani, Yulianti Oenang, Ali Akbar Septiandri, James Jaya, Kaustubh D. Dhole, Arie Ardiyanti Suryani, Rifki Afina Putri, Dan Su, Keith Stevens, Made Nindyatama Nityasya, Muhammad Farid Adilazuarda, Ryan Ignatius, Ryandito Diandaru, Tiezheng Yu, Vito Ghifari, Wenliang Dai, Yan Xu, Dyah Damapuspita, Cuk Tho, Ichwanul Muslim Karo Karo, Tirana Noor Fatyanosa, Ziwei Ji, Pascale Fung, Graham Neubig, Timothy Baldwin, Sebastian Ruder, Herry Sujaini, Sakriani Sakti, and Ayu Purwarianti. 2022. Nusacrowd: Open source initiative for indonesian NLP resources. *CoRR*, abs/2212.09648.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

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, Loïc 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. *CoRR*, abs/2207.04672.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2023. *Ethnologue: Languages of the World*, 26 edition. SIL International, Dallas, Texas.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *CoRR*, abs/2302.07856.

John Hewitt, Daphne Ippolito, Brendan Callahan, Reno Kriz, Derry Tanti Wijaya, and Chris Callison-Burch. 2018. Learning translations via images with a massively multilingual image dataset. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2566–2576.

Ahmad Fathan Hidayatullah, Siwi Cahyaningtyas, and Rheza Daffa Pamungkas. 2020. Attention-based cnn-bilstm for dialect identification on javanese text. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pages 317–324.

Alexander Jones and Derry Tanti Wijaya. 2021. Majority voting with bidirectional pre-translation for bitext retrieval. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 46–59.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Muhammed Kocyigit, Jiho Lee, and Derry Wijaya. 2022. Better quality estimation for low resource corpus mining. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 533–543.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Fajri Koto and Ikhwan Koto. 2020. Towards computational linguistics in Minangkabau language: Studies on sentiment analysis and machine translation. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 138–148, Hanoi, Vietnam. Association for Computational Linguistics.

Garry Kuwanto, Afra Feyza Akyürek, Isidora Chara Tourni, Siyang Li, Alexander Gregory Jones, and Derry Wijaya. 2021. Low-resource machine translation training curriculum fit for low-resource languages.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics.

Yingzhou Lu, Minjie Shen, Huazheng Wang, and Wenqi Wei. 2023. Machine learning for synthetic data generation: A review.

Oddy Virgantara Putra, Fathin Muhammad Wasmanson, Triana Harmini, and Shoffin Nahwa Utama. 2020. Sundanese twitter dataset for emotion classification. In *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pages 391–395.

Mohammad Sadegh Rasooli, Chris Callison-Burch, and Derry Tanti Wijaya. 2021. "wikily" supervised neural translation tailored to cross-lingual tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1655–1670, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Herry Sujaini. 2020. Improving the role of language model in statistical machine translation (indonesian-javanese). *International Journal of Electrical and Computer Engineering*, 10(2):2102.

Arie Ardiyanti Suryani, Dwi Hendratmo Widyantoro, Ayu Purwarianti, and Yayat Sudaryat. 2015. Experiment on a phrase-based statistical machine translation using pos tag information for sundanese into indonesian. In *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6.

Zilu Tang and Derry Wijaya. 2022. Knowledge based template machine translation in low-resource setting. *arXiv preprint arXiv:2209.03554*.

C Tho, Y Heryadi, L Lukas, and A Wibowo. 2021. Code-mixed sentiment analysis of indonesian language and javanese language using lexicon based approach. *Journal of Physics: Conference Series*, 1869(1):012084.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Rui Wang, Haipeng Sun, Kehai Chen, Chenchen Ding, Masao Utiyama, and Eiichiro Sumita. 2019. English-Myanmar supervised and unsupervised NMT: NICT's machine translation systems at WAT-2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 90–93, Hong Kong, China. Association for Computational Linguistics.

Haryo Akbarianto Wibowo, Made Nindyatama Nityasya, Afra Feyza Akyürek, Suci Fitriany, Alham Fikri Aji, Radityo Eko Prasojo, and Derry Tanti Wijaya. 2021. IndoCollex: A testbed for morphological transformation of Indonesian colloquial words. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3170–3183, Online. Association for Computational Linguistics.

Derry Tanti Wijaya, Brendan Callahan, John Hewitt, Jie Gao, Xiao Ling, Marianna Apidianaki, and Chris Callison-Burch. 2017. Learning translations via matrix completion. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1452–1463.

Derry Tanti Wijaya and Tom Mitchell. 2016. Mapping verbs in different languages to knowledge base relations using web text as interlingua. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 818–827.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 843–857. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, and Pascale Fung. 2023. Nusax: Multilingual parallel sentiment dataset for 10 indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 815–834. Association for Computational Linguistics.

Wilson Wongso, David Samuel Setiawan, and Derwin Suhartono. 2021. Causal and masked language modeling of javanese language using transformer-based architectures. In *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 1–7.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.

Bing Zhao and Stephan Vogel. 2002. Adaptive parallel sentences mining from web bilingual news collection.

In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 745–748. IEEE.

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

A Monolingual Training Data Breakdown

Lang	wikidumps	cc100	imdb-jv	jadi-ide	su-emot	Total	Filter
jv	382K	1.41M	100K	16K	0	1.91M	1.6M
su	220K	387K	0	0	2K	610K	550K
min	282K	0	0	0	0	282K	282K
ban	60K	0	0	0	0	60K	60K

Table 4: Monolingual training data breakdown. **Lang** denotes the language identifier, **Total** denotes the total monolingual sentences for each language, and **Filter** denotes how many monolingual sentences remain after filtering, as listed in our Methodology section.

The monolingual data of local languages in Indonesia are obtained from multiple sources, including parsed wikidumps, cc100 (Conneau et al., 2020), imdb-jv (Wongso et al., 2021), jadi-ide (Hidayatullah et al., 2020), and su-emot (Putra et al., 2020). Excluding wikidumps monolingual data, which was taken in December of 2022, all of these sources are obtained from the compilation done by Cahyawijaya et al. (2022) and was taken in January of 2023. The breakdown for these monolingual data of local languages in Indonesia is found in Table 4

For the monolingual data of the Indonesian language, we use the Indo4B curated dataset (Wilie et al., 2020). Excluding the data obtained from Twitter, the number of monolingual Indonesian sentences is 201 million. No sentences were filtered out due to the high quality of the dataset.

B Parallel Training Data Breakdown

Lang	su-id	min-nlp	code-mixed	bible	nusantara	nusax	Total	Filter
jv	0	0	977	7958	6000	1000	15935	14395
su	3616	0	0	7957	1699	1000	14272	13269
min	0	16371	0	0	0	1000	17371	17260
ban	0	0	0	0	0	1000	1000	997

Table 5: Parallel training data breakdown of the language pair Indonesia and the local language denoted by **Lang** (i.e., the entry **jv** list how much parallel data of the pair **ind, jv** are obtained from each source). **Total** denotes total parallel sentences for each pair of Indonesian and local languages, and **Filter** denotes how much remains after filtering, as listed in our Methodology section.

As with our monolingual data breakdown, all of our parallel data were obtained from the NusaCrowd repository in January 2023. The datasets we use include su-id (Suryani et al., 2015), min-nlp (Koto and Koto, 2020), code-mixed (Tho et al., 2021), bible (Cahyawijaya et al., 2021), nusantara (Sujaini, 2020), and nusax (Winata et al., 2023).

C Ablation of Different Methods in Generating Parallel Data

Model	id→jv	jv→id
text-davinci-003	17.95	26.26
gpt-3.5-turbo	19.10	30.19

Table 6: Zero-shot translation spm200BLEU score of generative AIs on the FLORES200 test set. Results indicate that gpt-3.5-turbo performs significantly more than text-davinci-003 on zero-shot translation.

We performed exploratory experiments regarding different methods of generating parallel data. As mentioned in our methodology, we define two main approaches to generating synthetic data: (1) Generating parallel data using generative AI and (2) Translating monolingual data using an already trained model.

The models we use to generate the parallel data in approach (1) are gpt-3.5-turbo and davinci-text-003. We limit our exploratory experiment to the **id↔jv** translation direction. First, we compare the zero-shot translation performance of these models on the FLORES200 test set, where gpt-3.5-turbo achieved a considerably higher spm200BLEU score. The full breakdown is available in Table 6. We give each model

Model	Approach	id→jv	jv→id
Baseline	-	12.99	12.29
text-davinci-003	zero-shot	14.86	16.88
	ten-shot	14.40	16.82
gpt-3.5-turbo	zero-shot	14.95	**17.10**
	ten-shot	**15.04**	16.87

Table 7: Performance of NMT systems trained from scratch using a **Semisup** paradigm when the original parallel data is mixed with synthetically generated data from gpt-3.5-turbo or text-davinci-003. The first row is the baseline model performance. All BLEU scores are from XLM's automated BLEU scoring. Bolded entries indicate the model with the best performance.

the prompt to generate these parallel data: **"Generate a long parallel sentence in SRC and TGT"**. Our internal experiments show that without the keyword **"long"**, the model will generate short and simple parallel sentences consisting of regularly occurring words.

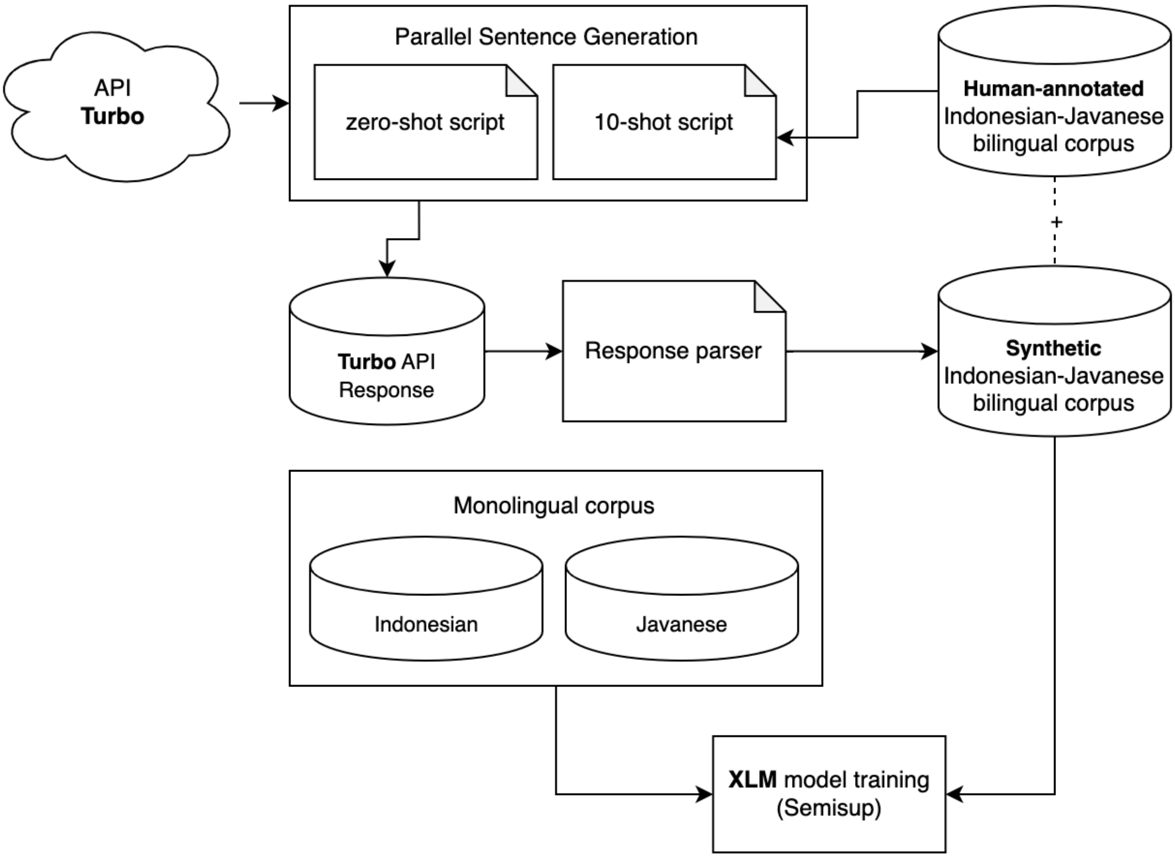


Figure 2: Illustration of using LLMs to generate synthetic parallel data. First tested on **id-jv** language pair, we use the same pipeline to generate synthetic parallel data for the other language pair.

We conduct these experiments using two approaches: zero-shot generation and ten-shot generation. We give the model the prompt above without additional context in zero-shot generation. We then parse the text that has been generated and split it into Indonesian sentences and Javanese sentences. In the ten-shot generation, we sample 10 parallel sentences in our original training data to feed it as examples to the model. Figure 2 illustrates this process. The impact these generated synthetic data have on training is found in Table 7. These performances align with the benchmark results in Table 6, where gpt-3.5-turbo is better at both translation directions than text-davinci-003. The results in Table 7 show that zero-shot generation of gpt-3.5-turbo creates parallel data with the most positive impact on NMT system training. The results shown in Table 7 indicate that few-shot may not have a considerable difference in performance compared to zero-shot for LLMs on translation tasks.

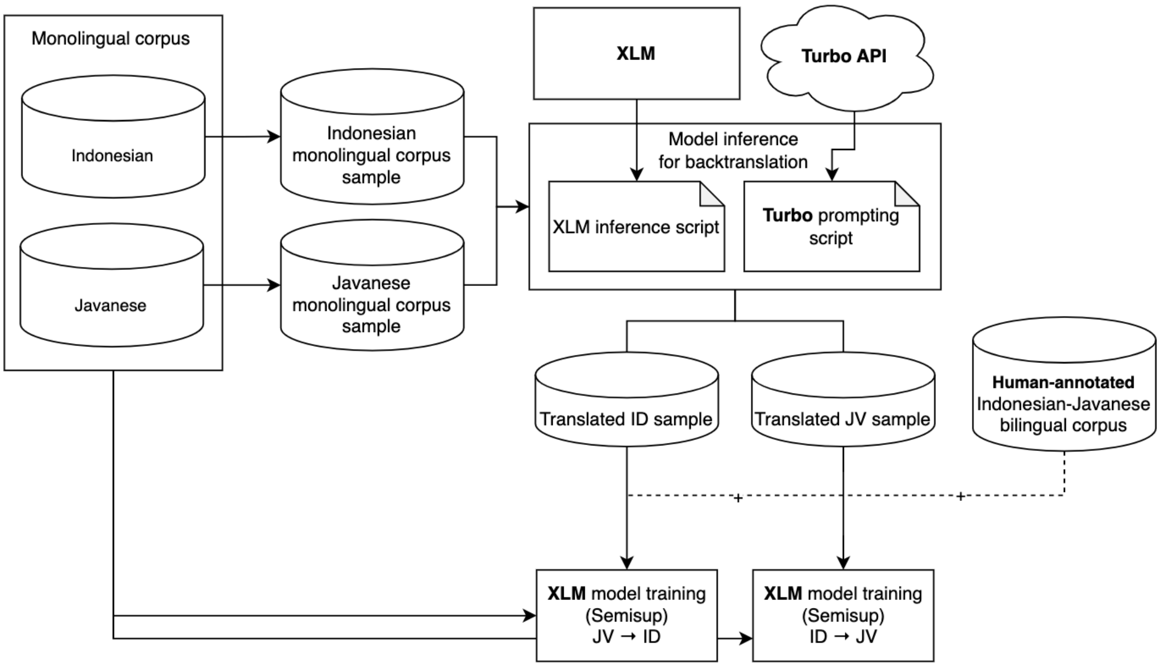

Monolingual corpus
Indonesian
Javanese
Indonesian monolingual corpus sample
Javanese monolingual corpus sample
XLM
Turbo API
Model inference for backtranslation
XLM inference script
Turbo prompting script
Translated ID sample
Translated JV sample
Human-annotated Indonesian-Javanese bilingual corpus
XLM model training (Semisup) JV → ID
XLM model training (Semisup) ID → JV

Figure 3: Illustration of translating monolingual data using an already trained model. In this work, we choose gpt-3.5-turbo and our NMT model trained on id-jv language pair using **Scratch** approach and **Semisup** paradigm.

Method	id→jv	jv→id
BT (baseline XLM)	10.35	12.29
BT (gpt-3.5)	13.22	13.80

Table 8: Comparison of additional data generation techniques. BT is for Back translation, in which we sample sentences from external monolingual corpora and translate them using the model indicated in the parentheses.

Besides prompting gpt-3.5 to generate parallel sentences directly, we also compared it with generating additional data from translating existing monolingual datasets. We use gpt-3.5 and the baseline XLM model to translate Wikipedia monolingual sentences. gpt-3.5-turbo is used instead of text-davinci-003 based on the results of the experiments shown in table 6. Our findings show that additional data from translating monolingual corpus using the baseline XLM model does not yield any significant performance increase or even hurts it, as shown in Table 8, whereas monolingual corpus translated using gpt-3.5 yields over 1 BLEU score on the Javanese to Indonesian translation direction.

However, this increase is modest compared to the results shown by directly generating parallel sentences from gpt-3.5-turbo as additional parallel data. Therefore, we move forward with approach (1). We apply the same procedure to the remaining language pairs: Sundanese, Minangkabau, and Balinese. Synthetic data generation is a promising research avenue in which both approaches (1) and (2) should still be included.

D Hyperparameters

In this work, there are a total of ten combinations of training approaches and paradigms which are:

- **Scratch** - **Unsup**
- **Scratch** - **Semisup**
- **Scratch**$_{AUG}$ - **Semisup**
- **PreXL** - Pre-training Language Model

- **PreXL** - **Unsup**
- **PreXL** - **Semisup**
- **CodeXL** - Pre-trained Language Model
- **CodeXL** - **Unsup**
- **CodeXL** - **Semisup**
- **CodeXL**$_{AUG}$ - Semisup

Across all of these models, nine hyperparameters are shared among them:

- `emb_dim`: 1024
- `n_layers`: 6
- `n_heads`: 8
- `dropout`: 0.1
- `attention_dropout`: 0.1
- `gelu_activation`: true
- `batch_size`: 32
- `bptt`: 256
- `epoch_size`: 200000

Models that are training using the **Unsup** training paradigm sets the hyperparameter **lambda_ae** to ***0:1,100000:0.1,300000:0***.

The hyperparameter **max_vocab** is set to ***200000*** for all models trained except models in **CodeXL** - Pre-trained Language Model, which uses the default values.

The hyperparameter **optimizer** is set to ***adam_inverse_sqrt, beta1=0.9, beta2=0.98, lr=0.0001*** for all models trained except models in **CodeXL** - Pre-trained Language Model and "**CodeXL** - **Unsup**" which set the value to ***adam, lr=0.0001***.

The hyperparameter **tokens_per_batch** is set to ***2000*** for all models excluding all models in **CodeXL** and **Scratch**$_{AUG}$ which set the value to ***4000***.