LexC-Gen: Generating Data for Extremely Low-Resource Languages with Large Language Models and Bilingual Lexicons

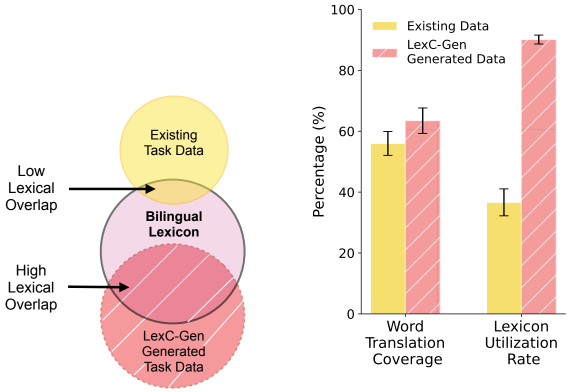
Zheng-Xin Yong[1] **Cristina Menghini**[2] **Stephen H. Bach**[1]
[1]Department of Computer Science, Brown University
[2]Data Science Institute, Brown University
{contact.yong,cristina_menghini,stephen_bach}@brown.edu

Abstract

Data scarcity in low-resource languages can be addressed with word-to-word translations from labeled task data in high-resource languages using bilingual lexicons. However, bilingual lexicons often have limited lexical overlap with task data, which results in poor translation coverage and lexicon utilization. We propose lexicon-conditioned data generation (LexC-Gen), a method that generates low-resource-language classification task data at scale. Specifically, LexC-Gen first uses high-resource-language words from bilingual lexicons to generate lexicon-compatible task data, and then it translates them into low-resource languages with bilingual lexicons via word translation. Across 17 extremely low-resource languages, LexC-Gen generated data is competitive with expert-translated gold data, and yields on average 5.6 and 8.9 points improvement over existing lexicon-based word translation methods on sentiment analysis and topic classification tasks respectively. Through ablation study, we show that conditioning on bilingual lexicons is the key component of LexC-Gen. LexC-Gen serves as a potential solution to close the performance gap between open-source multilingual models such as BLOOMZ and state-of-the-art commercial models like GPT-4o on low-resource-language tasks.

1 Introduction

Extremely low-resource languages do not have any labeled data and are thereby considered the "Left-Behinds" in NLP language technology development (Joshi et al., 2020; Mabokela et al., 2022; Robinson et al., 2023). Nonetheless, many of them have *bilingual lexicons* resources, which are usually the first product of language documentation (Meara, 1993; Schreuder and Weltens, 1993; Kroll and Ma, 2017). Bilingual lexicons are dictionaries that map words from one language to their translations in another languages, and they cover more than 5000 languages around the world (Wang et al., 2022; Koto et al., 2024).



(a) Intuition of LexC-Gen.

(b) Lexical overlap between lexicons and data.

Figure 1: We observe data-lexicon mismatch (i.e., low lexical overlap) between existing task data and bilingual lexicons (Figure 1a). LexC-Gen addresses the issue by generating data using words from lexicons so the data will have more words translated (i.e., higher word translation coverage) and higher lexicon utilization rate (Figure 1b).

Previous work uses bilingual lexicons to directly translate labeled data from high-resource languages to low-resource languages through word-for-word substitution (Wang et al., 2022; Jones et al., 2023, inter alia). However, we argue that it is ineffective because of *data-lexicon mismatch*. Often, the words in the *existing task data*—readily available labeled data in high-resource languages for a target task, e.g., sentiment analysis or topic classification—have low lexical overlap with the words in the task-agnostic bilingual lexicons, as shown in Figure 1. This mismatch not only results in many words remain untranslated, but also causes entries in the bilingual lexicon, which possibly contain useful semantic information for downstream tasks, missing from the translated dataset.

In this work, we introduce **LexC-Gen**,[1] which is a **lex**icon-**c**onditioned data **gen**eration method, to

[1] pronounced as lek-see-jen

Findings of the Association for Computational Linguistics: EACL 2024, pages 13990–14009
November 12-16, 2024 ©2024 Association for Computational Linguistics

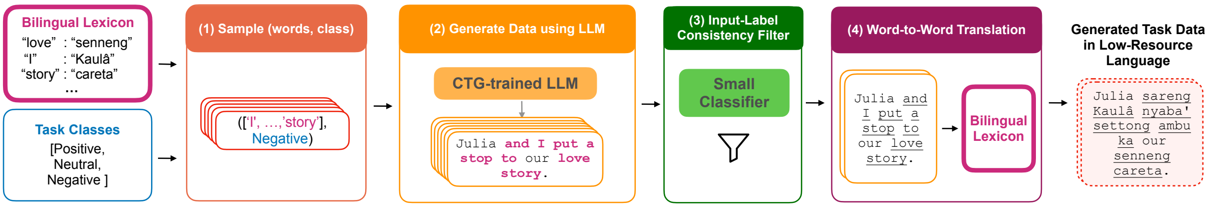

Bilingual Lexicon
"love" : "senneng"
"I" : "Kaulâ"
"story" : "careta"
...
Task Classes
[Positive, Neutral, Negative]
(1) Sample (words, class)
(['I', ...,'story'], Negative)
(2) Generate Data using LLM
CTG-trained LLM
Julia and I put a stop to our love story.
(3) Input-Label Consistency Filter
Small Classifier
(4) Word-to-Word Translation
Julia and I put a stop to our love story.
Bilingual Lexicon
Generated Task Data in Low-Resource Language
Julia sareng Kaulâ nyaba' settong ambu ka our senneng careta.

Figure 2: **LexC-Gen** Given a bilingual lexicon and the set of classes for a classification task, (1) we randomly sample the class label and a set of words from bilingual lexicon, for as many instances we desire to generate. (2) We use these pairs to build the prompts to query CTG-trained LLM (Figure 3) and generate the task data in high-resource language. (3) Then, we train a task classifier on existing task data to filter generated data and ensure input-label consistency. (4) After filtering, we apply word-to-word translation with the bilingual lexicon following prior work (Wang et al., 2022). Finally we get the synthetic task data for the target low-resource language, which is used to finetune task classifier.

mitigate data-lexicon mismatch through synthetic data generation. Specifically, we train LLMs to generate task data using words from bilingual lexicons, so the data have a higher lexical overlap with the lexicons. This results in better word translation coverage and lexicon utilization rate (Figure 1). We also propose a quality-control method that checks for input-label consistency to filter out poor-quality generated data.

We evaluated LexC-Gen across 17 extremely low-resource languages on sentiment analysis and topic classification tasks. We found that finetuning classifiers on LexC-Gen generated data improves on average 5.6 and 8.9 points in accuracy respectively over word-translated existing training data (Wang et al., 2022). Surprisingly, finetuning on LexC-Gen word-translated data even matches the performance of finetuning on *gold data* in the target language curated by native speakers or professional translators. We show that lexicon-conditioning is the critical success factor of LexC-Gen.

Finally, we discuss how LexC-Gen helps close the performance gap of open-source LLMs in low-resource-language tasks. We show that instead of zero-shot or few-shot prompting, it is better to use them to generate training data with LexC-Gen. The data generation process is cost-effective, and the permissive nature of the models allows generated data to be made open access for further research and building systems for extremely low-resource languages, which benefits multilingual NLP progress for these data-scarce languages.

Our contributions can be summarized as follows:

1. We present LexC-Gen, a method that conditions LLMs on bilingual lexicons to generate low-resource-language task data to address *data-lexicon mismatch* problem.
2. We demonstrate that training on word-translated task data can match training on *gold data* for extremely low-resource-languages.
3. Our extensive ablation study on LexC-Gen shows that simply scaling up generated task data is *insufficient*. Lexicon-conditioning is necessary to maximize lexical overlap between task data and bilingual lexicons.

2 Related Work

Generating task data with LLMs LLM-powered data generation is a recent promising area of research that enables cost-effective collection of diverse task data with minimal human labor (Honovich et al., 2023; Radharapu et al., 2023; Wang et al., 2023; Nayak et al., 2023; Yehudai et al., 2024). Nonetheless, this line of work has been underexplored in a multilingual setting. Whitehouse et al. (2023) demonstrated that GPT-4's generated multilingual training data for commonsense reasoning task in mid-/high-resource languages can improve cross-lingual performance. However, language coverage of LLMs and translation models are significantly smaller than lexicons (Wang et al., 2022; Bapna et al., 2022; Koto et al., 2024). Instead, we use LLMs to generate task data that maximize lexical overlap with bilingual lexicons for translations, and we show that our synthetic data can improve NLU semantic task performance in extremely low-resource languages.

Lexicon-based cross-lingual data augmentation Lexicon-based augmentation creates data for low-resource languages by swapping words in high-resource-language data with their dictionary word translations in bilingual lexicons. This is useful for low-resource languages that cannot be readily translated by translation models/APIs with limited language coverage. Prior work has demonstrated their effectiveness across a wide range of NLP tasks, such as machine translation (Streiter and Iomdin, 2000; Ramesh and Sankaranarayanan, 2018; Thompson et al., 2019; Kumar et al., 2022; Jones et al., 2023), sequence labeling (Scherrer and Sagot, 2013; Mayhew et al., 2017; Wang et al., 2022), sentiment classification (Rasooli et al., 2018; Ali et al., 2021; Mohammed and Prasad, 2023), and topic classification (Song et al., 2019). However, many lexicon-based data augmentation strategies for semantic tasks in low-resource languages rely on domain-specific lexicons (Das and Bandyopadhyay, 2010; Buechel et al., 2016; Mohammed and Prasad, 2023; Koto et al., 2024), and performance-wise they still fall short of gold training data collected in the target low-resource language (Rasooli et al., 2018; Koto et al., 2024). Our method LexC-Gen not only works with domain-agnostic bilingual lexicons, but also demonstrates competitive performance with gold training data on sentiment analysis and topic classification tasks across many low-resource languages.

3 LexC-Gen

We aim to generate data for classification tasks in a low-resource language L, given access to (1) labeled task data $\mathcal{T}_H$ with C classes in a high-resource language H, (2) a bilingual lexicon D_H^L that maps words from H to L, and (3) an LLM supporting H.

LexC-Gen uses these inputs to generate labeled task data $\widetilde{\mathcal{T}}_L$ in low-resource language. Our key idea is to prompt the LLM to generate task data using high-resource-language words from bilingual lexicons in order to create task data that have a higher lexical overlap with those bilingual lexicons (Figure 1a), and thus can be more effectively translated into L. In the following, we describe the steps to obtain $\widetilde{\mathcal{T}}_L$. For readability, we refer to D_H^L as D.

3.1 Sample Lexicon Words and Class Label

First, we randomly sample a set W_H of high-resource-language words w_H from D and a class label c. This corresponds to step (1) in Figure 2. The goal is to prompt our LLM to generate task inputs of class c using as many words from W_H as possible.

3.2 Generate Data with LLM Trained with Controlled-Text Generation (CTG)

Next, we prompt an LLM to generate high-resource-language task data $\widetilde{\mathcal{T}}_{H|D}$ conditioned on the bilingual lexicon. This is step (2) in Figure 2. However, because open-access instruction-tuned LLMs such as BLOOMZ (Muennighoff et al., 2023) are not finetuned for this purpose, we carry out controlled text generation (CTG) training of LLMs (Zhang et al., 2023; Zhou et al., 2023b) to create CTG-trained LLM.

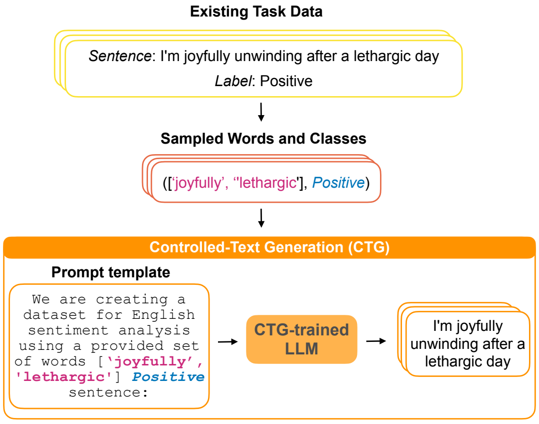

Existing Task Data
Sentence: I'm joyfully unwinding after a lethargic day
Label: Positive
Sampled Words and Classes
(['joyfully', "lethargic'], Positive)
Controlled-Text Generation (CTG)
Prompt template
We are creating a dataset for English sentiment analysis using a provided set of words ['joyfully', 'lethargic'] Positive sentence:
CTG-trained LLM
I'm joyfully unwinding after a lethargic day

Figure 3: **Controlled-Text Generation (CTG) training.** This figure shows the pipeline for the LLM finetuning for CTG. We construct the training data starting from the existing labeled task data $\mathcal{T}_H$. From each instance t_H, we sample without replacement a set of words W_H and associate it to class c. This information is plugged into the prompt template, and it is used to finetune an LLM that generates sentences conditioned on c and W_H.

CTG Training We construct CTG training data from existing task data $\mathcal{T}_H$. Each instance $t_H \in \mathcal{T}_H$ consists of a pair of text x_H and task label c. We randomly sample a variable number of word tokens w_H uniformly at random without repetition from x_H to create W_H. Then, we format the CTG training data using the prompt template in Figure 3, so that the LLM learns to generate task input $\tilde{x}_{H|c,W_H}$ conditioned on c and W_H.

CTG training is data-efficient. We found that generating only a single CTG training example per each $t_H \in \mathcal{T}_H$ is already sufficient to instruction-tune the model. Specifically, our CTG training data consists of 500 and 701 instances for our sentiment analysis and topic classification tasks respectively.

Task Data Generation After CTG training, we prompt the LLM reusing the template in Figure 3, but now we use lexicon words with random task class labels from Section 3.1. We can now generate synthetic high-resource-language task data $\widetilde{\mathcal{T}}_{H|D}$ at scale conditioned on bilingual lexicons.

Methods	**#data**	**ace**	**ban**	**bbc**	**bjn**	**bug**	**mad**	**min**	**Avg**
Zero-shot/Few-shot prompting									
BLOOMZ-7.1.B	0	47.0	50.5	43.0	49.5	38.5	48.0	52.5	47.0
Aya-101-13B	0	58.8	59.2	48.1	82.8	35.9	48.4	77.9	58.7
Aya-101-13B (few-shot)	5	60.8	**62.6**	53.0	**83.9**	45.7	53.9	**79.9**	62.8
GPT-4o	0	75.3	81.3	65.8	83.8	51.5	74.0	85.3	73.8
Cross-lingual zero-shot									
Existing Task Data (en)	500	56.8	60.2	51.1	63.3	45.8	56.0	57.7	55.8
Word translation									
Existing Task Data (T)	500	63.6	58.3	55.8	66.4	57.7	59.3	71.6	61.8
+ Existing Task Data (en)	1000	67.8	62.4	60.4	66.3	56.7	62.4	75.1	64.4
+ Label Distillation (Wang et al., 2022)	1000	58.8	52.9	45.7	58.8	43.9	56.8	68.7	55.1
LexC-Gen-1K (T)	∼ 370	42.4	47.1	49.6	53.9	43.5	42.3	44.3	46.2
+ Existing Task Data (en)	∼ 870	67.8	62.4	60.4	66.3	56.7	62.4	75.1	64.4
LexC-Gen-10K (T)	∼ 3.7K	66.6	67.1	61.0	72.3	57.3	61.2	70.7	65.2
+ Existing Task Data (en)	∼ 4.2K	68.2	67.0	62.8	71.4	58.5	57.9	70.3	65.2
LexC-Gen-100K (T)	∼ 37K	70.0	71.5	65.1	73.4	63.7	**69.9**	76.5	70.0
+ Existing Task Data (en)	∼ 38K	**70.7**	71.4	**67.8**	74.6	**65.8**	**69.9**	76.9	**71.0**
Gold Translations	500	72.1	71.6	68.6	72.8	68.1	66.7	77.3	71.0

Table 1: Sentiment analysis accuracy on 7 Indonesian extremely low-resource local languages in the NusaX dataset (Winata et al., 2023b). (T) indicates word-translated data, and (en) refers to the existing task data in English. The terms -1K, -10K and -100K refer to the size of training data generated by LexC-Gen before filtering. We **bold** the best result and underline the second-best. We report results averaged over 5 seeds of classifier finetuning.

3.3 Input-Label Consistency Filter

To ensure high-quality data, we apply an input-label consistency filter after data generation to reduce training noise from labeling errors. For instance, CTG-trained LLM may generate a sentence with negative sentiment even though the specified task label c is positive sentiment in the input prompt (Figure 3). Therefore, we finetune a small classifier mBERT on the same existing task data $\mathcal{T}_H$ and use it to relabel $\widetilde{\mathcal{T}}_{H|L}$. Then, we filter out all data instances where the classifier's prediction does not match the generated input-label pairs.

At this point (step (3) in Figure 2), we have high-quality lexicon-compatible task data in language H that allows for better word-to-word translation into language L by using D.

3.4 Word-to-Word Translation into Low-Resource Languages

Finally, we carry out word-to-word translation following the procedures in prior work (Wang et al., 2022; Jones et al., 2023). We use D to substitute the high-resource-language words $w_H \in \widetilde{\mathcal{T}}_{H|D}$ with their low-resource-language word translation w_L, thus creating $\widetilde{\mathcal{T}}_L$. We randomly sample w_L if w_H has multiple possible translations and keep w_H as is in $\widetilde{\mathcal{T}}_{H|D}$ if there is no translation for it in D. After we obtain the synthetic cross-lingual task data $\widetilde{\mathcal{T}}_L$, we use it as training data to finetune a classifier for the target task in the low-resource-language.

4 Experimental Setup

We compare LexC-Gen against baselines and gold translations on sentiment analysis and topic classification tasks. We describe the task datasets in Section 4.1, how we instantiate LexC-Gen in Section 4.2, and our baselines as well as gold translations in Section 4.3.

4.1 Tasks and Datasets

We evaluate LexC-Gen on sentiment analysis and topic classification tasks across 17 low-resource languages. The task datasets contain *gold training data* that are curated with translations by native speakers or professional translators. Detailed information for the tasks and languages can be found in Appendix B.

Sentiment analysis We use the NusaX sentiment analysis dataset (Winata et al., 2023b) developed for Indonesian low-resource languages. The dataset has 3 sentiment labels: positive, neutral, and negative. In our setup, we evaluate LexC-Gen on 7 languages that also exist in the Gatitos lexicon.

Topic classification SIB-200 (Adelani et al., 2023) is a topic classification benchmark that cov-

Methods	#data	bam	ewe	fij	grn	lin	lus	sag	tso	tum	twi	Avg
Zero-/Few-shot prompting												
BLOOMZ-7.1.B	0	41.7	34.3	35.3	41.7	42.2	38.7	36.8	41.7	40.2	41.7	39.4
Aya-101-13B	0	36.8	39.1	50.9	48.8	52.4	43.7	40.2	54.1	50.0	37.7	45.4
Aya-101-13B (few-shot)	5	42.2	46.1	60.4	55.1	59.7	48.2	49.4	**56.2**	**57.5**	43.8	51.9
GPT-4o	0	58.1	56.2	63.9	75.8	69.4	65.3	57.8	57.2	59.8	64.8	67.7
Cross-lingual zero-shot												
Existing Task Data (en)	701	29.6	32.5	42.5	57.7	42.0	49.9	37.6	39.6	40.3	40.7	41.2
Word translation												
Existing Task Data (T)	701	40.2	41.4	49.1	63.9	52.3	61.8	46.7	39.1	42.5	54.9	49.2
+ Existing Task Data (en)	1402	42.5	41.4	47.8	67.2	55.9	63.4	47.9	40.0	43.4	56.4	50.6
+ Label Distillation (Wang et al., 2022)	1402	37.5	33.1	41.9	59.0	37.8	56.5	38.5	42.1	41.2	35.0	42.3
LexC-Gen-1K (T)	∼ 220	22.9	37.8	40.2	50.1	45.0	52.5	40.9	29.2	37.6	42.1	39.8
+ Existing Task Data (en)	∼ 920	36.5	41.2	45.3	68.3	53.0	61.9	49.1	37.1	39.0	53.7	48.5
LexC-Gen-10K (T)	∼ 2.2K	38.5	40.5	51.4	67.1	57.6	64.1	55.3	41.1	42.6	55.1	51.3
+ Existing Task Data (en)	∼ 2.9K	33.8	42.6	51.3	67.1	59.3	64.8	53.8	43.8	43.2	54.3	51.4
LexC-Gen-100K (T)	∼ 22K	44.0	**51.1**	**70.2**	**74.3**	**67.4**	**69.3**	**61.0**	42.2	50.9	64.9	**59.5**
+ Existing Task Data (en)	∼ 23K	**46.2**	47.6	68.0	73.0	67.2	68.9	57.0	42.6	53.0	**65.8**	58.9
Gold Translations	701	54.9	53.0	61.7	71.2	64.6	68.4	60.7	55.9	63.4	62.2	61.6

Table 2: Topic classification accuracy for 10 worst-performing languages in the SIB-200 dataset (Adelani et al., 2023). We follow the schema defined in Table 1.

ers 200 languages and 7 topic categories. We evaluate LexC-Gen on the *10 worst-performing languages* that we found to have the largest performance gap between gold translations and the word translation baseline (Wang et al., 2022).

4.2 LexC-Gen Instantiation

LLM We use the BLOOMZ model (Muennighoff et al., 2023) with 7.1 billion parameters (BLOOMZ-7.1B) as our initial instruction-tuned LLM. This allows us to compare performance between its zero-shot prompting and its usage with LexC-Gen.

Bilingual lexicons We choose Gatitos bilingual lexicons (Jones et al., 2023) to translate the generated English data into low-resource languages. Gatitos includes English entries such as frequent English words, numbers, and time, and they are translated into 170 extremely low-resource languages. Gatitos have been manually reviewed, so its entries have higher quality than other bilingual lexicons such as Panlex (Kamholz et al., 2014).

Generated task data We first use LexC-Gen to generate English datasets with 1K, 10K, and 100K instances, to which we refer as LexC-Gen-1K, -10K, and -100K, before filtering out mismatched input-label pairs. The effective data size after filtering with input-label consistency checking is between 20% and 40% of the generated task data. Then, we use Gatitos lexicons (Jones et al., 2023) to translate them into low-resource languages.

Training and data generation with LLM We provide further training and inference details of LexC-Gen in Appendix C. We also showcase examples of the generated data for sentiment analysis in Table 3 and for topic classification in Table 6.

Task finetuning We finetune pretrained mBERT[2] with classification heads on LexC-Gen generated low-resource-language data for sentiment analysis and topic classification tasks evaluation (further details are in Appendix D).

4.3 Baselines

We compare LexC-Gen against (1) **zero-shot prompting** with BLOOMZ-7.1B, Aya-101-13B (Üstün et al., 2024) and GPT-4o;[3] (2) **few-shot prompting** with Aya-101-13B using five in-context learning examples; (3) **cross-lingual zero-shot transfer** where mBERT is finetuned on English training data and evaluated on low-resource-language test data; (4) **word translation** (Wang et al., 2022) where mBERT is finetuned on the data that are translated from the English training data via word-substitution with the same bilingual lexicon Gatitos (Jones et al., 2023); (5) **gold translations**

[2] `bert-base-multilingual-cased` model.

[3] We used the latest version `gpt-4o-2024-05-13`. See Appendix E for more details.

Generated Text	Sentiment
ulon 'm reusam leumeeh ngeun hek , ulee sikula papeun tuleh member . Hike trails , ta'jub jamek let man keun keu lon . (I'm feeling weak and tired, principal board member. Hike trails, wonderful plural pursuit but not for me.)	Negative
Please , peutamah nyan pre uteun handbook jadwal keulayi keu umum ureung umum , nyan 's jareung hadiah lam nyan areusip (Please, extend the free forest handbook schedule for general public, it's hardly present in the archive)	Neutral
Wonderful , trang ngeun mangat , superior guna , tajam ngeun carong , ngeun nyan barang nakeuh superb . ulon nasihat meujuang toke 's ho jak keu nyan , nyan 's saboh konfiden peuningkat . (Wonderful, bright and comfortable, superior service, sharp and smart, and the package is superb. I advise struggling entrepreneur's to go for it, it's a confidence booster.)	Positive

Table 3: Text samples of generated sentiment analysis data in Acehnese language by LexC-Gen. The English words that remain untranslated are underlined. The bracketed English text is the originally generated text by LexC-Gen in Section 3.2 before being tokenized and translated with the bilingual lexicons in Section 3.4.

where mBERT is finetuned on expert-translated task training data in the target low-resource language (see Section 4.1)

We implement the word translation baseline by referring to the state-of-the-art method (Wang et al., 2022). Here, we do not adapt the pretrained mBERT before task finetuning for fair comparison. We follow the protocol by Wang et al. (2022) and report the result where we also combine word-translated data with English training data ("+ Existing Task Data (en)") and perform *label distillation*—a technique that uses a classifier (mBERT in our case) trained on existing task data to relabel the translated data.

5 Results and Analysis

5.1 LexC-Gen improves over open-source LLMs and direct word translation

LexC-Gen outperforms prompting open-source models, such as BLOOMZ and Aya-101, and word translation baselines in both sentiment analysis (Table 1) and topic classification tasks (Table 2). In sentiment analysis, finetuning classifiers on the mixture of LexC-Gen-100K (100K generated data instances that are filtered down to around 37K instances) and existing English task data improves over the cross-lingual zero-shot baseline by 15.2 percentage points and word translation baseline by 6.6 points. In topic classification, LexC-Gen-100K yields improvement of 18.3 points over the cross-lingual zero-shot baseline and 8.9 points over the word translation baseline. The accuracy gain from adding existing English data reduces from LexC-Gen-1K to LexC-Gen-100K because the English data are dominated by the substantially larger size of generated data (see more discussion in Appendix I).

While the commercially available model GPT-4o yields the best performance—even surpassing classifiers trained on gold data—it is unclear whether the evaluation data has been seen during training. Furthermore, the release of GPT-4o is subsequent to our work. In contrast, our evaluation tasks of NusaX and SIB-200 are not part of the training of open-source models BLOOMZ-7.1B and Aya-101-13B (Workshop et al., 2022; Singh et al., 2024; Üstün et al., 2024). Our results reveal the performance gap in these open-source models. For instance, zero-shot prompting with BLOOMZ-7.1B is the weakest baseline (Table 1 and Table 2). However, using it in LexC-Gen to generate task data (i.e., LexC-Gen-100K) can achieve state-of-the-art performance. Our results suggest that, **for applying open-source LLMs to low-resource-language tasks, it is best to leverage them to generate training data at scale** instead of prompting them directly in zero-shot or few-shot settings.

LexC-Gen-100K improves over baselines because first, it improves the word translation coverage of data instances (Figure 1b left) so there are fewer undesirable artifacts of untranslated words in high-resource languages. Second, it significantly increases the lexicon utilization rate (Figure 1b right and Section 5.4), which allows more low-resource-language words from the lexicon to be present in the task data so the task classifier can associate task labels with the semantic information carried by these words.

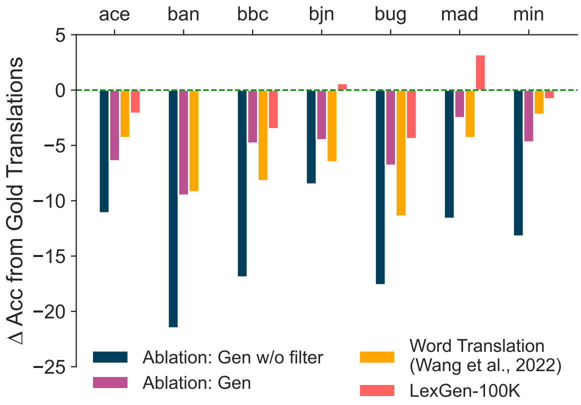


Figure 4: Ablation study of lexicon-conditioning in LexC-Gen-100K on sentiment analysis. The plot shows that accuracy difference against finetuning with gold translations (green dotted line).

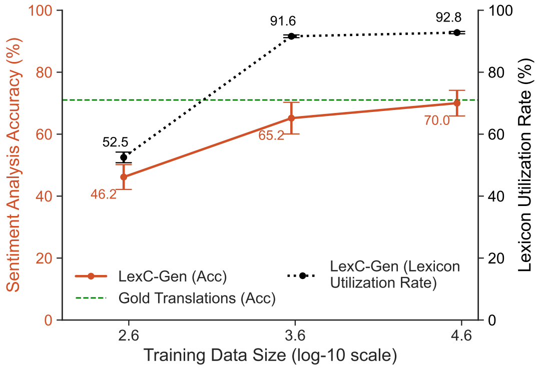


Figure 5: Sentiment analysis accuracy (red solid line, left y-axis) and lexicon utilization rate (blue dotted line, right y-axis) against the size of LexC-Gen training task data in log10-scale.

5.2 LexC-Gen is competitive with gold translations

Table 1 and Table 2 show that finetuning classifiers on LexC-Gen-100K generated cross-lingual data is competitive with training on expert-translated data for many low-resource languages. Our findings also generalize to larger task classifiers, such as XLMR-base and XLMR-large (Conneau et al., 2020) (see Figure 7 in Section 5.6). Our result is surprising because LexC-Gen generated data still use English syntax with SVO word order. Yet, LexC-Gen still works for languages with different word orders, such as Balinese (ban) and Mizo (lus) with OSV word order and Toba batak (bbc) with VOS word order.

One possible explanation is that solving sentiment analysis and topic classification tasks relies more on semantic information than syntactic information. Because of the larger word translation coverage and extremely high lexicon utilization rate (Figure 1b), LexC-Gen generated data at scale contain sufficient semantic information in low-resource languages for classifiers to learn the task. Nonetheless, it requires a much larger LexC-Gen dataset to match gold translations performance. LexC-Gen data (after filtering) are around 75× and 30× the size of gold translations as shown in Table 1 and Table 2 for sentiment analysis and topic classification tasks respectively.

5.3 Lexicon-conditioning is crucial for strong task performance

Figure 4 shows that using words from lexicons to generate task data (i.e., lexicon-conditioning) is necessary for matching gold translations performance. Ablating away lexicon-conditioning and quality control ("Gen w/o filter") has the worst performance— it even underperforms the word translation baseline (Wang et al., 2022) on 500 existing task data samples for sentiment analysis. Even with quality control from Section 3.4, scaling data generation without lexicon conditioning ("Gen") still performs worse than LexC-Gen-100K. This is due to low lexical overlap between the data and bilingual lexicons. "Gen" data have poorer lexicon utilization rate, as it only covers 62.5% of low-resource-language words in the bilingual lexicon. In contrast, LexC-Gen-100K covers 92.8% words. We refer our readers to Appendix F for further details of our ablation study.

5.4 Scaling generated data increases lexicon utilization rate

Figure 5 shows that scaling up the data generation process improves the utilization rate of bilingual lexicons, which is the total proportion of low-resource-language words in bilingual lexicons appearing in the translated dataset, because LexC-Gen uses more words from lexicons to generate task data. We observe that as lexicon utilization rate improves, sentiment analysis accuracy increases. This is because there is more semantic information for classifiers to learn the downstream tasks in the target language. We also obtain a similar graph with the topic classification task (see Appendix Figure 9). Scaling is enabled by the generative nature of LexC-Gen, as opposed to previous approaches constrained to the quantity of labeled task data in high-resource languages.

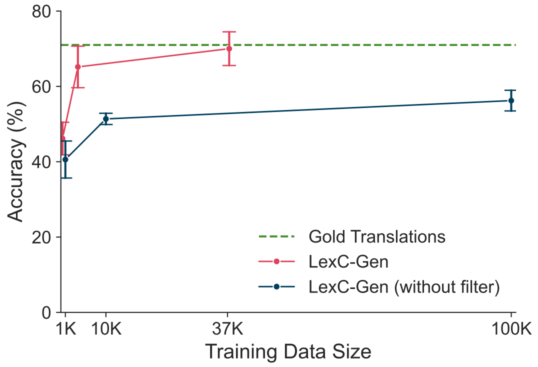


Figure 6: Ablation of input-label consistency filter on LexC-Gen generated data for sentiment analysis.

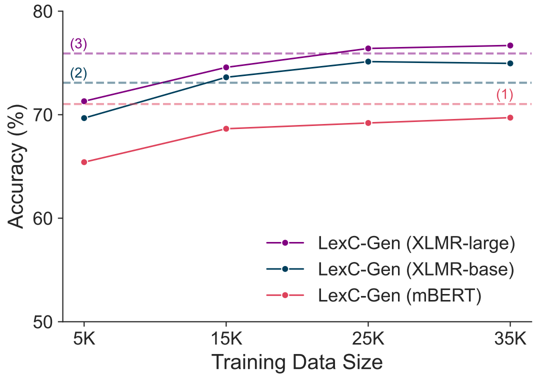


Figure 7: Sentiment analysis accuracy on the NusaX dataset (averaged across all 7 languages) with different task classifiers. The dotted lines (1), (2), and (3) represent the accuracy for mBERT, XLMR-base and XLMR-large classifiers when trained on gold translations respectively.

5.5 Quality control reduces training data size and boosts performance

Figure 6 shows that applying input-label consistency filter as data quality control not only reduces the size of the generated training data by two-third, which results in 3 times faster finetuning of the task classifier, but also increases the task performance from 56.2 points (ablation of quality control at 100K generated data) to 70.0 points (37K generated data after quality control filtering), which even matches the performance of finetuning on gold translations. Our findings align with prior work with English data (Zhou et al., 2023a) that shows that optimizing for data quality results in more significant gains than simply scaling up data quantity.

Quality control with a classifier trained on existing task data is effective for LexC-Gen, but not for label distillation in Wang et al.'s (2022) word translation baseline (Table 1 and Table 2). There are two possible reasons. First, label distillation uses the classifier trained on high-resource-language data to relabel translated data in low-resource languages. This cross-lingual transfer may introduce errors in the classifier's predictions, as opposed to LexC-Gen's relabeling in the same high-resource language. Second, LexC-Gen offers *stricter* quality control by discarding all instances with mismatched labels between the classifier and LLMs, thus improving task performance (see Figure 11 in Appendix H).

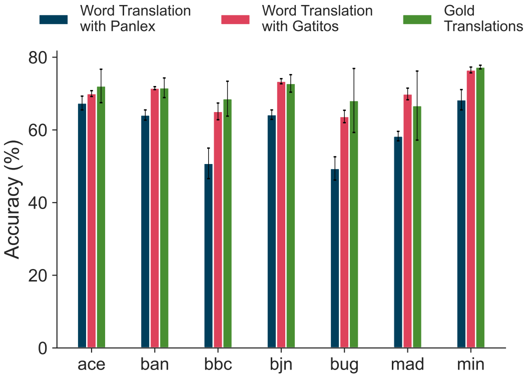


Figure 8: Sentiment analysis accuracy on NusaX dataset between word translation of LexC-Gen generated data with Gatitos (Jones et al., 2023) and Panlex (Kamholz et al., 2014).

5.6 LexC-Gen generalizes to larger classifiers

Figure 7 breaks down the LexC-Gen generated data size required for task classifiers of different sizes—mBERT (Devlin et al., 2019) has 172 million parameters, XLMR-base (Conneau et al., 2020) has 270 million parameters, and XLMR-large has 550 million parameters—to match gold translations performance. First, we observe that larger task classifier size requires *less data* to achieve same accuracy. For instance, XLMR-large already exceeds accuracy of 70 points with 5K LexC-Gen data but mBERT requires 35K LexC-Gen data to reach the same accuracy.

However, to match gold performance, the amount of LexC-Gen data does not correlate with the size of classifiers. We find that XLMR-base matches gold performance at around 15K, as opposed to mBERT at around 35K, but XLMR-large requires around 10K more LexC-Gen data than XLMR-base to be as competitive as gold translations.

5.7 Choice of lexicons matters

We compare the usage of Gatitos (Jones et al., 2023) against Panlex (Kamholz et al., 2014) as the lexicons for LexC-Gen. Figure 8 shows that translating LexC-Gen generated data with Gatitos outperforms translating with Panlex on NusaX. One reason is that Panlex has a smaller lexicon size than Gatitos, as for all seven languages in NusaX, Panlex only has around 840 entries, but Gatitos has around 4271 entries. Therefore, LexC-Gen data have a poorer word translation coverage with Panlex. Furthermore, while the data source of Gatitos is not detailed by Jones et al. (2023), the authors described that Gatitos entries are manually reviewed. In other words, the word translations with Gatitos are of higher quality.

6 Discussion

Application in resource-scarce scenarios LexC-Gen addresses the resource-scarce scenarios faced by extremely low-resource languages that lack labeled data. We show that we can leverage their existing-yet-scarce lexical resources such as Gatitos (Jones et al., 2023), which only contains around a few thousand of lexicon entries for common words or phrases, to generate labeled task data. Furthermore, LexC-Gen is a practical solution to *data-lexicon mismatch* problem, as it does not require linguists to build task-specific lexicons, such as multilingual sentiment lexicons (Chen and Skiena, 2014), or practitioners to collect labeled task data in low-resource languages.

Open-source models Current open-source models like BLOOMZ (Muennighoff et al., 2023) and Aya-101 (Üstün et al., 2024) fail to close the performance gap against GPT-4o and gold translation baseline. Our work bridges the gap—we show that using them to generate training data improves performance over direct zero-shot or few-shot prompting and can match training classifiers on human-labeled data. Furthermore, due to the permissive nature of the models, their generated data can be used for proprietary or public research for broader multilingual applications.

Effectiveness of lexicon-conditioned generation We shows that task-agnostic bilingual lexicons like Gatitos (Jones et al., 2023) al contain *sufficient* semantic information for sentiment analysis and topic classification in extremely low-resource languages. However, it requires a high degree of lexical overlap between task data and lexicon to include the information in the translated data (Figure 1a). We also found that lexicon size and quality are important. Using Gatitos lexicons (Jones et al., 2023) for LexC-Gen outperforms using Panlex (Kamholz et al., 2014) because the former contains more entries and is higher in quality.

LexC-Gen differs from prior work on lexically constrained text generation (Hokamp and Liu, 2017; Post and Vilar, 2018; Hu et al., 2019). We introduce an additional step of CTG training so LLMs can learn to generate natural text that both maximizes lexicon usage and aligns with class labels. This step allows LexC-Gen to outperform lexically constrained decoding (see Appendix G).

Cost-effectiveness LexC-Gen relies on the CTG-trained LLM that follows the prompt instruction of generating task data using a set of given words. Our CTG training of open-source LLMs only depends on high-resource-language task data, and is independent of low-resource languages and bilingual lexicons. In other words, once an LLM is CTG-trained, researchers can *reuse* it with different bilingual lexicons to generate data for various low-resource languages on the same task *without retraining*. Furthermore, LexC-Gen only takes less than a day to generate 100K data samples on one V100 GPU.

Bilingual lexicon induction (BLI) We analyze the generated data and discover that on average 34% of the high-resource-language words cannot be found in Gatitos and thus cannot be translated. This leaves room for improvement with BLI to expand the word coverage of bilingual lexicons (Nasution et al., 2016; Irvine and Callison-Burch, 2017; Bafna et al., 2024), so that more words can be translated into low-resource languages. Nonetheless, given that LexC-Gen already matches gold performance, we leave enhancing LexC-Gen with BLI to future work.

7 Conclusion

We propose LexC-Gen to generate low-resource-language task data for sentiment analysis and topic classification tasks. Finetuning on our generated data can match gold data that are difficult to collect. Given that LexC-Gen is a practical solution, we hope it alleviates the severe data scarcity problem of low-resource languages and accelerates NLP progress in these long-tail languages.

Limitations

Word ambiguity In our word-to-word translation, we follow the protocol of prior work (Wang et al., 2022) and randomly choose a word translation if a particular word is mapped to multiple translations. In other words, we do not disambiguate word translations in low-resource languages because the low-resource-language words existing in lexicons do not come with linguistic information (such as parts-of-speech tags) or context (such as example sentences) that are necessary for word sense disambiguation (Navigli, 2009). Therefore, our word translations may introduce errors in the translated task data. Future work could expand the entries in bilingual lexicons to incorporate linguistic or contextual information to enable word sense disambiguation and improve the quality of the translated data in low-resource languages.

Syntax mismatch Since LexC-Gen is based on word-to-word translation, it suffers the inherent limitation that the syntax of its generated word-translated sentences remains unchanged and therefore might not match that of low-resource languages. Nonetheless, we have shown that despite this limitation, LexC-Gen still improves performance significantly in semantic tasks such as sentiment analysis and topic classification for languages with different word orders. This suggests that LexC-Gen is a viable solution for semantic tasks when in-language training data are extremely difficult to collect for low-resource languages. Future work should explore syntactical transformation of LexC-Gen's synthetic data to better align with low-resource languages for tasks, such as machine translation and named entity recognition, that heavily rely on syntactic information.

Tasks We experimented LexC-Gen on sentiment analysis and topic classification tasks, both of which are NLU tasks that low-resource languages are still lagging behind high-resource languages (Winata et al., 2023b; Adelani et al., 2023). We acknowledge that future work is warranted to explore the potentials and limitations of LexC-Gen on other NLU tasks that (1) require sensitivity to semantic complexity at the sentence level, such as common sense reasoning and natural language inference, or (2) syntax information, such as named entity recognition and information retrieval.

Source language In our experiments, we follow prior work (Jones et al., 2023; Wang et al., 2022) and generate low-resource-language task data from English task data using English-based Gatitos bilingual lexicons (Jones et al., 2023). Future work should explore extending LexC-Gen beyond English and generating task data in high-resource languages that are more related to the low-resource languages than English language. It would also be interesting to explore if BLOOMZ or other open-access LLMs are capable in terms of controlled-text generation abilities for non-English languages.

Broader Impacts and Ethical Considerations

Since our work addresses the training data scarcity problem of extremely low-resource languages (Joshi et al., 2020; Yong et al., 2023; Singh et al., 2024, inter alia), we foresee adoption and further research of our methods by NLP practitioners for tackling other NLU semantic tasks. Since our approach works well with LLMs with permissive licenses, it is possible that the generated task data are widely distributed for NLP applications in many different low-resource languages.

One potential risk of synthetic data is model collapse (Shumailov et al., 2023) where synthetic data cause the tails of the original data distribution disappear. Here, our work focuses on synthetic data for long-tail languages. We want to emphasize that LexC-Gen's generated cross-lingual training data *are not* substitute for natural in-language data. Our work actually encourages more human investment in low-resource languages in terms of lexicon curation and task data collection. We not only demonstrate that high-quality bilingual lexicons are effective in improving semantic task performance, but also show that gold translations in the target low-resource language require less data to achieve strong task performance.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and Annie En-Shiun Lee. 2023. Sib-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects.

Wazir Ali, Naveed Ali, Yong Dai, Jay Kumar, Saifullah Tumrani, and Zenglin Xu. 2021. Creating and evaluating resources for sentiment analysis in the low-resource language: Sindhi. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 188–194, Online. Association for Computational Linguistics.

Niyati Bafna, Cristina España-Bonet, Josef van Genabith, Benoît Sagot, and Rachel Bawden. 2024. When your Cousin has the Right Connections: Unsupervised Bilingual Lexicon Induction for Related Data-Imbalanced Languages. In *LREC-Coling 2024 - Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, Proceedings of the The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, Torino, Italy.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

Sven Buechel, Johannes Hellrich, and Udo Hahn. 2016. Feelings from the Past—Adapting affective lexicons for historical emotion analysis. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 54–61, Osaka, Japan. The COLING 2016 Organizing Committee.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389, Baltimore, Maryland. 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.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Amitava Das and Sivaji Bandyopadhyay. 2010. Sentiwordnet for bangla. *Knowledge Sharing Event-4: Task*, 2:1–8.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

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

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Alexander Jones, Isaac Caswell, Orhan Firat, and Ishank Saxena. 2023. GATITOS: Using a new multilingual lexicon for low-resource machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 371–405, Singapore. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

David Kamholz, Jonathan Pool, and Susan M Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *LREC*, pages 3145–3150.

Fajri Koto, Tilman Beck, Zeerak Talat, Iryna Gurevych, and Timothy Baldwin. 2024. Zero-shot sentiment analysis in low-resource languages using a multilingual sentiment lexicon.

Judith F Kroll and Fengyang Ma. 2017. The bilingual lexicon. *The handbook of psycholinguistics*, pages 294–319.

Nalin Kumar, Deepak Kumar, and Subhankar Mishra. 2022. Dict-nmt: Bilingual dictionary based nmt for extremely low resource languages. *arXiv preprint arXiv:2206.04439*.

Koena Ronny Mabokela, Turgay Celik, and Mpho Raborife. 2022. Multilingual sentiment analysis for under-resourced languages: A systematic review of the landscape. *IEEE Access*.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545, Copenhagen, Denmark. Association for Computational Linguistics.

Paul Meara. 1993. The bilingual lexicon and the teaching of vocabulary. *The bilingual lexicon*, pages 279–297.

Idi Mohammed and Rajesh Prasad. 2023. Building lexicon-based sentiment analysis model for low-resource languages. *MethodsX*, 11:102460.

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.

Arbi Haza Nasution, Yohei Murakami, and Toru Ishida. 2016. Constraint-based bilingual lexicon induction for closely related languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3291–3298, Portorož, Slovenia. European Language Resources Association (ELRA).

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. 2023. Learning to generate instructions to adapt language models to new tasks. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Ayu Purwarianti and Ida Ayu Putu Ari Crisdayanti. 2019. Improving bi-lstm performance for indonesian sentiment analysis using paragraph vector. In *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–5. IEEE.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Bhaktipriya Radharapu, Kevin Robinson, Lora Aroyo, and Preethi Lahoti. 2023. Aart: Ai-assisted red-teaming with diverse data generation for new llm-powered applications. *arXiv preprint arXiv:2311.08592*.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. 2018. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Mohammad Sadegh Rasooli, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. 2018. Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32:143–165.

Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for high- (but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.

Yves Scherrer and Benoît Sagot. 2013. Lexicon induction and part-of-speech tagging of non-resourced languages without any bilingual resources. In *RANLP Workshop on Adaptation of language resources and tools for closely related languages and language variants*.

Robert Schreuder and Bert Weltens. 1993. *The bilingual lexicon*, volume 6. John Benjamins Publishing.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. Model dementia: Generated data makes models forget. *arXiv e-prints*, pages arXiv–2305.

Eduardo Marín Silva. 2021. On the 1978 version of the african reference alphabet.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh

Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning.

Yangqiu Song, Shyam Upadhyay, Haoruo Peng, Stephen Mayhew, and Dan Roth. 2019. Toward any-language zero-shot topic classification of textual documents. *Artificial Intelligence*, 274:133–150.

Oliver Streiter and Leonid L Iomdin. 2000. Learning lessons from bilingual corpora: Benefits for machine translation. *International journal of corpus linguistics*, 5(2):199–230.

Brian Thompson, Rebecca Knowles, Xuan Zhang, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. HABLex: Human annotated bilingual lexicons for experiments in machine translation. 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 1382–1387, Hong Kong, China. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

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

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

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*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Yifan Gao, and Daniel Preotiuc-Pietro. 2023a. Efficient zero-shot cross-lingual inference via retrieval. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 93–104, Nusa Dua, Bali. 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, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023b. 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*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *arXiv preprint arXiv:2401.14367*.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina. 2023. BLOOM+1: Adding language support to BLOOM for zero-shot prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. Controlled text generation with natural language instructions. In *Proceedings of the 40th International Conference on Machine Learning*, volume

202 of *Proceedings of Machine Learning Research*, pages 42602–42613. PMLR.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model.

A Public Release

The **project page** for LexC-Gen is on `https://batsresearch.github.io/lexcgen/`.
The **code** repository for LexC-Gen is `https://github.com/BatsResearch/LexC-Gen`.
All **data** artifacts are on `https://github.com/BatsResearch/LexC-Gen-Data-Archive`.

B Tasks and Languages

We evaluated LexC-Gen on sentiment analysis and topic classification tasks across 17 extremely low-resource languages. All of them are classified as 0 or 1 in Joshi et al.'s (2020) taxonomy. Appendix B shows the language information of all the languages covered by our evaluation tasks. The datasets we use here are for research purposes.

For NusaX sentiment analysis dataset (Winata et al., 2023b), the authors employ two expert annotators who are native speakers of each local language to translate text from Indonesian sentiment analysis dataset (Purwarianti and Crisdayanti, 2019; Wilie et al., 2020) while maintaining the sentence's sentiment polarity, preserving entities, and maintaining the complete information content of the original text. The dataset has 500 train, 100 validation, and 400 test examples for each language.

Our baseline BLOOMZ has only been exposed to 5 out of 17 languages, which are Bambara, Lingala, Tsonga, Tumbuka, and Twi. These languages are in the topic classification tasks.

For SIB-200 topic classification dataset (Adelani et al., 2023), it is constructed using the translations from FLORES-200 (Costa-jussà et al., 2022), a multi-way parallel corpus that are curated with professional translators. The authors annotated the English portion of the Flores-200 dataset and extend the topic classification labels to the remaining 204 languages covered in FLORES-200. The dataset contains 701 training examples, 99 validation examples, and 204 test examples for each language for each language.

C CTG Training and Data Generation Details

CTG training of LLMs We construct the CTG training dataset, which have 500 and 701 English instances respectively, for sentiment analysis and topic classification following CTG training part of Section 3.2. Then, we finetune BLOOMZ-7.1B model (with permissive RAILS license) on a single V100 GPU using `BitsAndBytesConfig` and `LoraConfig` from `transformers` library for 4-bit QLoRA parameter-efficient finetuning (Dettmers et al., 2023). With 4-bit QLoRA, we can now finetune 7-billion parameter LLMs on commercially available GPUs without special setup (which otherwise would have been challenging as such finetuning would be restricted to GPUs with larger GPU memory such as A100 40GB GPUs.). We use the paged AdamW optimizer and set the learning rate to $2e^{-4}$, the sequence length to 1024, and the total effective training batch size to 1. We use the following hyperparameters for QLoRA adapters (Table 5).

We perform CTG training for 10 epochs and save the checkpoint every 500 steps. The entire CTG training can be finished within an hour on a single GPU.

Selection of CTG-trained LLM checkpoint After CTG training, we want to select the best model checkpoint that can maximize the usage of provided English word tokens when generating task data so the task data will have more lexical coverage with bilingual lexicons. Section 3.4. Specifically, we prompt the model to generate $\widetilde{T}_X$ input text and measure how well it uses tokens from $L_{w_X \sim D_X^Y}$ to generate text. The best checkpoint is the one that uses the most tokens. In practice, it is already sufficient to select the best checkpoint by evaluating only 200 generations per checkpoint.

In our search for the best generation hyperparameters, we found that either a low p or a low temperature (but not both at the same time) is the best for models to maximize the usage of provided tokens to generate text.

Data generation For each data instance generation, we randomly sample 10 high-resource-language (English) words from the bilingual lexicons and a class label to prompt the CTG-trained LLM, using the prompt template from Figure 3, to generate a maximum of 256 tokens. All these sampled words from lexicons do **not** come with

Languages	ISO Code	Task	Is seen?	Language Family	Subgrouping	Script	Word Order
Acehnese	ace	SA	✗	Austronesian	Malayo-Polynesian	Latin	SOV
Balinese	ban	SA	✗	Austronesian	Malayo-Polynesian	Latin	OVS
Toba batak	bbc	SA	✗	Austronesian	Malayo-Polynesian	Latin	VOS
Banjarese	bjn	SA	✗	Austronesian	Malayo-Polynesian	Latin	SVO
Buginese	bug	SA	✗	Austronesian	Malayo-Polynesian	Latin	VOS
Madurese	bug	SA	✗	Austronesian	Malayo-Polynesian	Latin	SVO
Minangkabau	min	SA	✓	Austronesian	Malayo-Polynesian	Latin	SVO
Bambara	bam	TC	✗	Niger-Congo	Mande	Latin	SOV
Ewe	ewe	TC	✗	Atlantic-Congo	Volta-Congo	Latin	SVO
Fijian	fij	TC	✗	Austronesian	Malayo-Polynesian	Latin	VOS
Guarani	grn	TC	✗	Tupian	Tupi–Guaran	Latin	SVO
Lingala	lin	TC	✗	Atlantic–Congo	Benue–Congo	Latin	SVO
Mizo	lus	TC	✗	Sino-Tibetan	Tibeto-Burman	Latin	OSV
Sango	sag	TC	✗	Atlantic–Congo	Ngbandi-based creole	Latin	SVO
Tsonga	tso	TC	✗	Atlantic–Congo	Volta–Congo	Latin	SVO
Tumbuka	tum	TC	✗	Atlantic–Congo	Volta–Congo	Latin	SVO
Twi	twi	TC	✗	Atlantic–Congo	Kwa	Latin	SVO

Table 4: Languages covered in our sentiment analysis (SA) and topic classification (TC) evaluation tasks. "Is seen?" refers to whether the language has been seen in pretraining of our mBERT task classifier. Note that while many African languages as well as Guarani language use Latin-based scripts, they have language-specific alphabets such as African reference alphabets (Silva, 2021) and Guarani alphabets (e.g., G̃/g̃).

Hyperparameters	**Values**
Dropout	0.1
α	16
r	64

Layers	`query_key_value,`
	`dense,`
	`dense_h_to_4h,`
	`dense_4h_to_h`

Table 5: Hyperparameters for QLoRA (Dettmers et al., 2023) finetuning for controlled text generation (CTG) training of LLMs.

linguistic information (such as parts-of-speech tags information) or task-related information (such as whether the words are topic or sentiment related). Following our findings before, we perform top-p sampling using $p = 0.1$ and temperature of 1 for data generation.

Input-label consistency filter We finetune mBERT classifier on our existing English task data in high-resource languages (English) following the setup described in Appendix D. On the English validation set (existing task data), it has 84.6 ± 0.7 and 86.6 ± 2.9 accuracy points for sentiment analysis and topic classification respectively. Then, we use the classifier to relabel the generated data and filter out instances where the classifier's labels do not match the original provided labels that are used to prompt LLMs to generate data in LexC-Gen.

Word-to-word translation After filtering the generated data, we tokenize the words using the English Stanza tokenizer (Qi et al., 2020) and then perform word-to-word substititon with the bilingual lexicon as described in Section 3.4. We follow Wang et al. (2022) and do not perform any lemmatization or stemming before word translation, as our preliminary experiments found that they introduce noises and harm task performance.

D Finetuning Task Classifiers

Task classifiers For both sentiment analysis and topic classification tasks, we finetune our mBERT classifier for 100 epochs in all setups with early stopping with patience of 3 evaluated on task validation sets. All finetuning runs took between 5 to 20 epochs to complete because of early stopping, allowing each run (even for on LexC-Gen's larger-scale generated task dataset) to be completed within 24 hours on a single V100 GPU. We use a batch size of 32, a learning rate of $1e^{-5}$, and the AdamW optimizer for classifier finetuning.

Task validation set To select the best task classifier for evaluation after finetuning on LexC-Gen generated training data, we use the validation set that is readily provided along with the task (instead of splitting our LexC-Gen generated data into train-validation data splits) and is word-translated. Specifically, we translate the English validation datasets with word-for-word substitution using bilingual lexicons and select the best classifier using the highest F1 score on the word-translated validation set. We also use this word-translated validation set for our word translation baseline (Wang et al., 2022). For cross-lingual zero-shot baseline, we use the readily available English task validation data.

E Zero-Shot/Few-Shot Prompting

BLOOMZ-7B1 and Aya-101-13B For zero-shot prompting with BLOOMZ-7B1 and Aya-101-13B, we use the prompts created for sentiment analysis and topic classification tasks in xP3 (Muennighoff et al., 2023) and take the average accuracy scores.

GPT-4o We use `gpt-4o-2024-05-13` and follow Adelani et al. (2023) for their zero-shot prompting template for the topic classification task: "Is this a piece of news regarding {{'science, technology, travel, politics, sports, health, entertainment, or geography'}}? {{INPUT}}" For sentiment analysis, we adapt the prompt to become "Does this sentence have {{'positive, negative, neutral'}} sentiment? {{INPUT}}"

F Ablation of Lexicon-Conditioning

Lexicon-conditioned generation refers to generating data using words from lexicons. In our ablation study in Section 5.3, we ablate away two components: lexicon-conditioning and quality control with input-label consistency filter.

Gen w/o filter This refers to generating data with LLM that only learns to generate task data in CTG. In other words, we remove the provided set of words in the prompt in Figure 3 when we perform CTG-training. In data generation, we do not provide words from lexicons, and we use high temperature and high p ($p = 0.9$) in top-p sampling so the CTG-trained LLM can generate diverse task data. After data generation, we did not perform any quality control filtering. This ablation setup measures the significance of both lexicon-conditioned generation and input-label consistency filter.

Gen This follows **Gen w/o filter** above but with filtering to ensure that the generated task data have matching labels and input text. This ablation setup measures the significance of lexicon-conditioned generation.

Controlled variables In both **Gen** and **Gen w/o filter**, we control the training data size by randomly sampling a subset of data so that they match the effective training dataset size of LexC-Gen-100K after input-label consistency filtering. Aside from removal of lexicon-conditioning prompt as described above and high p for sampling, the CTG training and data generation setups used for **Gen** and **Gen w/o filter** are the same as LexC-Gen-100K.

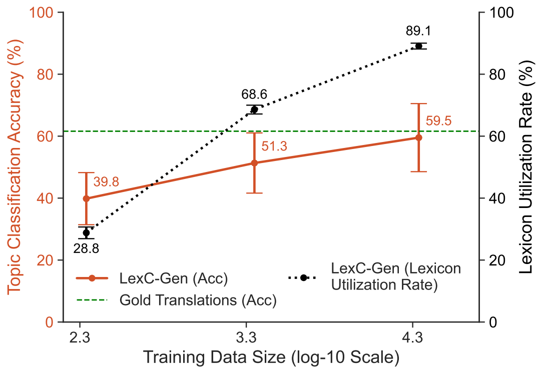


Figure 9: Topic classification accuracy (red, left y-axis) and lexicon utilization rate (blue, right y-axis) against the size of LexC-Gen task data in log10-scale.

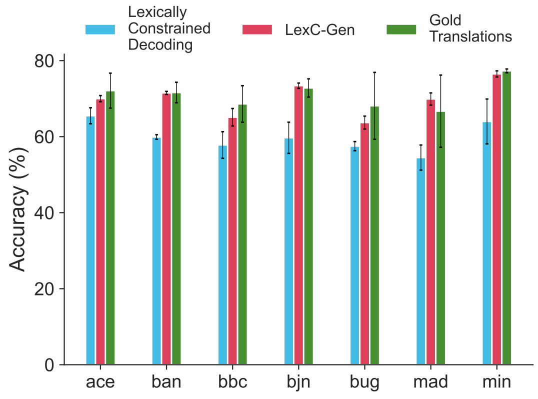


Figure 10: Sentiment analysis accuracy on NusaX dataset between lexically constained decoding (Post and Vilar, 2018; Hu et al., 2019) and LexC-Gen.

G Lexically Constrained Decoding

Lexically constrained decoding is an inference-time technique that enforces explicit word-/phrase-based constraints in generation (Hokamp and Liu, 2017; Post and Vilar, 2018; Hu et al., 2019) so that certain words and phrases will appear in output strings. We are curious if it can also create lexicon-compatible task data like LexC-Gen. We use out-of-the-box lexically constrained decoding method, implemented in the HuggingFace's `generation` function with `force_words_ids`, to generate from BLOOMZ-7.1B model finetuned

Generated Text	Topic
Badminton y bi agodie mu de w he players fa di dwuma badges (fr rackets mu tennis) k b balls k mu bi sap . (Badminton is a game in which players use badges (called rackets in tennis) to hit balls into a net.)	Sports
The mptam mfikyifuo y located so no koko boro so no refugee camp ne serves s bi ahynsode firi no camp 's pere k kora no nkae firi no tragedy te ase ber a moving so . (The community garden is located on the hill above the refugee camp and serves as a symbol of the camp's struggle to keep the memory of the tragedy alive while moving on.)	Travel
Information visualization enne becomes bi akade k boa users te ase kuntann asm . (Information visualization then becomes a tool to help users understand complex information.)	Science/ Technology
Voters mu France b si gyinae mu bi referendum so June 15 s k ma kwan saa ara - sex civil unions . (Voters in France will decide in a referendum on June 15 whether to allow same-sex civil unions.)	Politics
aane , no awia aduane bu y ber bn nnipa k firi mu firi wn kwan k w bi ny nkmmdie , k y anigye firi , anaas embarrass obi . (Yeah, the lunch break is when people go out of their way to have a bad conversation, to make fun of, or embarrass someone.)	Entertainment
Benada 's nkaeb na y wie a bi ayarehw agyinatukuo firi nhwehwmu]ulconcluded a Mr. Garfield 's owuo na n aso k akwanhyia . (Tuesday's announcement was made after a medical board of inquiry concluded that Mr. Garfield's death was not due to accident.)	Health
Rarely y ahum surges , de w he y no san tene firi waves breaking adum no mpoano , duru no mpoano . (Rarely do storm surges, which are the return flow from waves breaking off the shore, reach the beach.)	Geography

Table 6: Text samples of generated topic classification data in Twi language by LexC-Gen. The English words that remain untranslated are underlined. The bracketed English text is the originally generated text by LexC-Gen in Section 3.2 before being tokenized and translated with the bilingual lexicons in Section 3.4.

only on controlled-text generation task with class label c (i.e., "Gen" models in Section 5.3) with beam size of 5. We apply the lexical constraint such that a random subset of 10 words tokens from bilingual lexicons will appear in the model's generations of task inputs given a class label. We generate 100K samples from lexically constrained decoding and apply the same input-label consistency filter.

Figure 10 shows that lexically constrained decoding underperforms LexC-Gen. Upon non-exhaustive inspection of the generated instances, we find that while lexically constrained decoding yields generations with high lexicon utilization rate, in many cases it simply join some lexicon tokens together in order to satisfy the lexical constraint, hence forming grammatically incorrect and unnatural sentences. This suggests that it is non-trivial to generate natural sentences using random and independent word tokens in inference time.

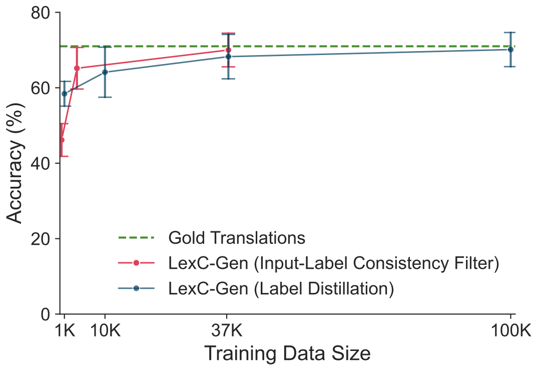


Figure 11: Relabeling all labels for generated data (i.e., label distillation (Wang et al., 2022)) as opposed to input-label consistency filter for LexC-Gen on sentiment analysis.

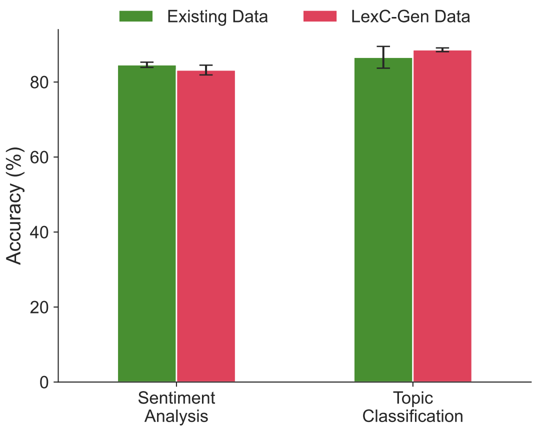


Figure 12: English language performance for sentiment analysis and topic classification.

H Label Distillation for LexC-Gen Generated Data

We extend the quality control study in Section 5.5 and compare LexC-Gen's input-label consistency filter against label distillation for LexC-Gen (Wang et al., 2022), where we use the mBERT classifier trained on existing English task data to *relabel* all LexC-Gen generated data. Since label distillation does not filter out poor-quality data instances, the generated data from LexC-Gen-1K, -10K and -100K remains the same. Therefore, for fair comparison against our state-of-the-art LexC-Gen-100K performance, we randomly sample data subsets from the relabeled 100K data to match the size of filtered LexC-Gen-100K training data at 37K samples.

Figure 11 shows that simply relabeling generated data (blue line) underperforms by input-label consistency filter (red line) at training data size of 37K. For label distillation to match the performance, we need 100K relabeled data, which is significantly more than filtered LexC-Gen data and thus incurs significant task finetuning costs. Therefore, input-label consistency filter is a better quality control method as it gives better task performance while reducing the training data size.

I Mixing in English task data helps for small-scale translated data

In both word translation baseline (Existing Task Data (T)) and LexC-Gen-1K with small-scale translated data, including existing English task data during classifier finetuning improves task performance substantially. For instance, in sentiment analysis, it yields 18.2 points performance gain for LexC-Gen-1K. However, at larger scales of data such as LexC-Gen-100K, mixing in English task data only gives marginal performance gain; for instance, 1 point average gain in the sentiment analysis task. This is because LexC-Gen-100K has around 37K training examples (after input-label consistency filtering), which dominate over the small-sized existing English task data with 500 examples. [4]

J Do Generated Data Help High-Resource Languages?

While our work is focusing on low-resource languages, we are interested in whether our LexC-Gen generated data in English can also help English tasks (that LexC-Gen is CTG-trained on). We compared filtered LexC-Gen-100K data and existing English data (which are the gold task data) for both sentiment analysis and topic classification tasks.

Figure 12 shows that for sentiment analysis, using existing data (which have 500 training examples) outperforms LexC-Gen data (which have around 37K examples) by average 1.4 points. On the other hand, for topic classification, LexC-Gen data (which have around 22K examples) outperforms existing data (which have 701 examples) by average 2.0 points. Similar to our findings with low-resource languages, LexC-Gen generated data are also as competitive as gold data for high-resource languages. However, the synthetic data do not bring significant performance gains in high-resource-language tasks where labeled data are readily available.

K Full Results for Larger Task Classifiers

We also report results with XLMR-base and XLMR-large task classifiers (Conneau et al., 2020) for sentiment analysis (Table 7 and Table 8) and topic classification (Table 9 and Table 10).

[4] In the following subsections, analysis of LexC-Gen does not include English existing task data.

Methods	#data	ace	ban	bbc	bjn	bug	mad	min	Avg
Zero-shot prompting									
BLOOMZ-7.1.B	0	47.0	50.5	43.0	49.5	38.5	48.0	52.5	47.0
Aya-101-13B	0	58.8	59.2	48.1	82.8	35.9	48.4	77.9	58.7
Aya-101-13B (few-shot)	5	60.8	**62.6**	53.0	**83.9**	45.7	53.9	**79.9**	62.8
GPT-4o	0	75.3	81.3	65.8	83.8	51.5	74.0	85.3	73.8
Cross-lingual zero-shot									
Existing Task Data (en)	500	54.3	55.4	40.0	66.1	38.0	50.0	68.9	53.2
DistFuse (Winata et al., 2023a)	500	65.5	70.5	65.3	75.3	58.0	67.3	73.5	67.9
Word translation									
Existing Task Data (T)	500	69.0	62.4	65.5	76.9	59.8	64.4	70.7	67.0
+ Existing Task Data (en)	1000	68.0	72.7	63.4	80.5	59.1	73.8	81.2	71.2
+ Label Distillation (Wang et al., 2022)	1000	63.1	66.4	58.4	73.0	44.2	67.8	80.1	64.7
LexC-Gen-1K (T)	∼ 370	38.4	38.0	38.3	38.9	38.3	38.2	39.2	38.5
+ Existing Task Data (en)	∼ 870	70.1	70.2	56.5	78.2	43.3	60.2	73.0	64.5
LexC-Gen-10K (T)	∼ 3.7K	70.4	70.0	59.8	78.2	61.7	67.8	79.0	69.6
+ Existing Task Data (en)	∼ 4.2K	70.6	71.2	61.9	79.3	62.7	68.0	79.3	70.4
LexC-Gen-100K (T)	∼ 37K	75.3	**77.7**	71.2	81.7	**68.3**	73.3	**81.8**	75.6
+ Existing Task Data (en)	∼ 38K	**75.6**	77.0	**73.0**	**81.8**	66.1	**75.2**	81.5	**75.7**
Gold Translations	500	73.9	75.8	64.2	76.1	68.2	71.8	78.8	72.7

Table 7: Sentiment analysis accuracy on 7 Indonesian extremely low-resource local languages in the NusaX dataset (Winata et al., 2023b) with XLMR-base classifier (Conneau et al., 2020). We follow the schema defined in Table 1. We also include the reported scores from another baseline DistFuse (Winata et al., 2023a) that uses cross-lingual retrieval to improve NusaX task performance.

Methods	#data	ace	ban	bbc	bjn	bug	mad	min	Avg
Zero-shot prompting									
BLOOMZ-7.1.B	0	47.0	50.5	43.0	49.5	38.5	48.0	52.5	47.0
Aya-101-13B	0	58.8	59.2	48.1	82.8	35.9	48.4	77.9	58.7
Aya-101-13B (few-shot)	5	60.8	**62.6**	53.0	**83.9**	45.7	53.9	**79.9**	62.8
GPT-4o	0	75.3	81.3	65.8	83.8	51.5	74.0	85.3	73.8
Cross-lingual zero-shot									
Existing Task Data (en)	500	65.8	71.4	39.6	78.4	35.2	61.5	81.8	62.0
Word translation									
Existing Task Data (T)	500	71.0	60.8	64.9	74.4	58.1	69.1	82.3	68.7
+ Existing Task Data (en)	1000	73.1	78.2	67.2	82.7	58.1	67.8	80.1	72.5
+ Label Distillation (Wang et al., 2022)	1000	65.4	70.9	70.9	73.4	45.6	71.1	77.8	67.9
LexC-Gen-1K (T)	∼ 370	38.2	38.5	43.1	40.4	39.0	38.2	42.6	40.0
+ Existing Task Data (en)	∼ 870	71.5	74.3	59.5	82.5	54.5	70.1	79.9	70.3
LexC-Gen-10K (T)	∼ 3.7K	68.0	69.9	68.3	81.8	61.8	67.3	83.2	71.5
+ Existing Task Data (en)	∼ 4.2K	68.3	77.2	63.9	83.9	60.3	70.3	**85.3**	72.7
LexC-Gen-100K (T)	∼ 37K	74.6	78.8	**73.2**	83.5	**68.3**	75.1	82.2	76.5
+ Existing Task Data (en)	∼ 38K	**75.9**	**79.1**	72.3	8**4.7**	67.1	**76.7**	84.2	**77.1**
Gold Translations	500	76.6	75.6	65.8	84.4	65.3	77.0	83.5	75.5

Table 8: Sentiment analysis accuracy on 7 Indonesian extremely low-resource local languages in the NusaX dataset (Winata et al., 2023b) with XLMR-large classifier (Conneau et al., 2020). We follow the schema defined in Table 1.

Methods	**#data**	**bam**	**ewe**	**fij**	**grn**	**lin**	**lus**	**sag**	**tso**	**tum**	**twi**	**Avg**
Zero-shot prompting												
BLOOMZ-7.1.B	0	41.7	34.3	35.3	41.7	42.2	38.7	36.8	41.7	40.2	41.7	39.4
Aya-101-13B	0	36.8	39.1	50.9	48.8	52.4	43.7	40.2	54.1	50.0	37.7	45.4
Aya-101-13B (few-shot)	5	42.2	46.1	60.4	55.1	59.7	48.2	49.4	**56.2**	**57.5**	43.8	51.9
GPT-4o	0	58.1	56.2	63.9	75.8	69.4	65.3	57.8	57.2	59.8	64.8	67.7
Cross-lingual zero-shot												
Existing Task Data (en)	701	33.1	38.4	35.6	57.2	42.1	59.3	42.0	36.7	35.2	43.1	42.3
Word translation												
Existing Task Data (T)	701	37.5	36.9	44.8	66.5	51.3	63.5	47.5	39.6	42.3	50.6	48.1
+ Existing Task Data (en)	1402	40.0	36.8	45.9	66.3	48.2	62.5	47.7	41.5	44.4	51.8	48.5
+ Label Distillation (Wang et al., 2022)	1402	37.5	22.5	40.4	62.5	44.4	60.4	45.3	41.1	43.2	37.9	43.5
LexC-Gen-1K (T)	∼ 220	17.8	27.9	29.4	34.8	31.0	24.9	29.8	28.6	29.2	29.8	28.3
+ Existing Task Data (en)	∼ 920	31.8	37.8	37.3	65.0	50.0	59.7	46.8	35.9	37.9	48.1	45.0
LexC-Gen-10K (T)	∼ 2.2K	39.3	40.3	50.0	64.2	55.9	66.5	55.0	41.4	46.5	54.9	51.4
+ Existing Task Data (en)	∼ 2.9K	36.9	42.4	50.6	67.2	55.9	64.8	54.6	39.8	46.4	53.9	51.2
LexC-Gen-100K (T)	∼ 22K	48.4	51.6	**62.5**	**73.0**	**68.0**	70.3	58.0	41.7	**53.7**	**62.7**	59.0
+ Existing Task Data (en)	∼ 23K	**48.6**	**53.6**	**62.5**	72.7	65.2	**72.8**	**60.3**	41.2	53.3	61.7	**59.2**
Gold Translations	701	31.2	53.7	38.1	68.6	63.1	69.5	56.7	44.8	56.5	58.0	54.0

Table 9: Topic classification accuracy for 10 worst-performing languages in the SIB-200 dataset (Adelani et al., 2023) with XLMR-base classifier (Conneau et al., 2020). We follow the schema defined in Table 1.

Methods	**#data**	**bam**	**ewe**	**fij**	**grn**	**lin**	**lus**	**sag**	**tso**	**tum**	**twi**	**Avg**
Zero-shot prompting												
BLOOMZ-7.1.B	0	41.7	34.3	35.3	41.7	42.2	38.7	36.8	41.7	40.2	41.7	39.4
Aya-101-13B	0	36.8	39.1	50.9	48.8	52.4	43.7	40.2	54.1	50.0	37.7	45.4
Aya-101-13B (few-shot)	5	42.2	46.1	60.4	55.1	59.7	48.2	49.4	**56.2**	**57.5**	43.8	51.9
GPT-4o	0	58.1	56.2	63.9	75.8	69.4	65.3	57.8	57.2	59.8	64.8	67.7
Cross-lingual zero-shot												
Existing Task Data (en)	701	29.6	27.2	32.1	63.6	39.9	56.0	41.6	38.3	41.6	43.1	41.3
Word translation												
Existing Task Data (T)	701	42.4	43.1	48.5	70.6	52.9	66.4	43.4	43.5	47.7	52.9	51.1
+ Existing Task Data (en)	1402	43.1	45.2	45.2	71.7	54.8	65.7	49.9	43.1	50.9	54.3	52.4
+ Label Distillation (Wang et al., 2022)	1402	37.9	27.8	42.9	64.6	43.5	58.9	48.3	42.6	48.8	39.5	45.5
LexC-Gen-1K (T)	∼ 220	23.5	32.4	33.9	47.1	35.3	44.7	34.1	27.2	33.0	26.2	33.7
+ Existing Task Data (en)	∼ 920	37.5	45.7	41.8	70.2	52.8	60.7	48.2	43.3	44.6	51.0	49.6
LexC-Gen-10K (T)	∼ 2.2K	43.2	46.6	53.3	68.1	59.1	68.1	50.6	46.2	55.5	53.2	54.4
+ Existing Task Data (en)	∼ 2.9K	37.5	44.3	51.7	69.2	57.7	68.1	49.6	42.4	51.3	58.2	53.0
LexC-Gen-100K (T)	∼ 22K	50.5	**54.6**	66.0	74.1	**67.5**	**70.7**	56.7	45.2	**56.2**	**62.8**	60.4
+ Existing Task Data (en)	∼ 23K	**52.4**	53.2	**67.4**	**76.8**	67.0	70.0	**57.3**	45.0	53.1	62.5	**60.5**
Gold Translations	701	50.6	60.9	58.3	73.1	64.1	68.2	62.5	48.4	60.0	65.8	61.2

Table 10: Topic classification accuracy for 10 worst-performing languages in the SIB-200 dataset (Adelani et al., 2023) with XLMR-large classifier (Conneau et al., 2020). We follow the schema defined in Table 1.