Text Filter Based on Automatically Acquired Vocabularies for Multilingual Machine Translation

Kenji Imamura and Masao Utiyama
National Institute of Information and Communications Technology
Hikari-dai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{kenji.imamura, mutiyama}@nict.go.jp

Abstract

In this paper, we propose a text filter designed to support multiple languages. The method simply aggregates vocabulary from a monolingual corpus and compares it against the input. Despite its simplicity, the approach proves highly effective in removing code-mixed text. When combined with existing language identification techniques, our method can enhance the purity of the corpus in the target language. Consequently, applying it to parallel corpora for machine translation has the potential to improve translation quality. Additionally, the proposed method supports the incremental addition of new languages without the need to retrain those already learned. This feature easily enables our method to be applied to low-resource languages.

1 Introduction

Multilingual models are becoming increasingly common because neural models can process multiple languages using a single model (Johnson et al., 2017). For example, encoder-based models include multilingual BERT (mBERT; Devlin et al., 2019) and XLM-RoBERTa (XLM-R; Conneau et al., 2020), while encoder-decoder models feature multilingual BART (including mBART-50; Liu et al., 2020; Tang et al., 2020). In addition, decoder-only models, which are commonly referred to as large language models (LLMs), such as Llama 3, Qwen, and GPT-oss also support multiple languages.

However, the multilingual corpora used to train these models are inherently noisy, as they are often acquired automatically from web sources. For instance, Briakou et al. (2023) reported that even in well-cleaned monolingual training data for LLMs, approximately 1.4% contained a mixture of other languages. To remove such undesirable data, filtering is essential. However, corpora often include languages that even the dataset creators cannot comprehend, and automatic processing is necessary.

Language identification is an effective approach for excluding non-target languages from corpora (see Section 2.2 for details). However, conventional methods often fail to remove texts that contain segments of other languages, which are referred to as code-mixed texts in this paper, when the proportion of those languages is relatively small.

In this paper, we propose a monolingual filtering method designed to support multiple languages. The proposed approach performs token-level identification in a straightforward manner by matching against automatically acquired vocabulary. We apply this method to parallel corpora that include alignment scores or have been pre-filtered using alternative techniques, and demonstrate that it improves machine translation quality.

Our proposed method is particularly effective in removing code-mixed texts. By using corpora with fewer code-mixed instances for model training, systems can generate more consistent language, thereby improving machine translation quality.

Furthermore, new languages can be added incrementally, as the approach only requires tokenizing monolingual corpora of the target language and aggregating tokens to construct the vocabulary. This eliminates the need to retrain previously learned languages. For low-resource languages, an independent filtering method is particularly valuable, since major language identifiers may not reliably support them.

The remainder of this paper is organized as follows. Section 2 introduces related work, including multilingual corpus filtering and language identification. Section 3 provides a detailed description of the proposed method. Section 4 presents experiments on parallel corpus filtering and machine translation evaluation to validate the effectiveness of the proposed approach. Finally, Section 5 concludes the paper.

Proceedings for the Ninth Workshop on Technologies for Machine Translation of Low Resource Languages (LoResMT 2026), pages 37–48

2 Related Work

2.1 Multilingual Corpus Filtering

Many multilingual corpora are collected from the web. Data obtained from sources other than Wikipedia are typically filtered using language identification and other techniques. In this section, we first summarize methods for filtering multilingual corpora.

2.1.1 Collection of Monolingual Corpora

CC-100 is a collection of monolingual corpora covering 100 languages (Conneau et al., 2020). It was constructed following the procedure used for building the CCNet corpus (Wenzek et al., 2020). The procedure can be summarized as follows:

1. The collected web pages are deduplicated at the paragraph level.

2. A fastText-based language identifier (Joulin et al., 2016; Bojanowski et al., 2017) is used to detect the language of each page, and pages with low identification scores are removed.

3. For 48 high-resource languages, language models are trained for each language to compute paragraph-level perplexity (PPL). Paragraphs with high PPL values are discarded, with thresholds determined individually for each language based on its PPL distribution.

To summarize, filtering is performed based on language identification scores at the page level and perplexity derived from language models at the paragraph level. Because both scores reflect the overall unit, texts containing code-mixing are not removed if the proportion of other languages is minimal.

2.1.2 Parallel Corpora

CCMatrix (Schwenk et al., 2021b) and WikiMatrix (Schwenk et al., 2021a), both multilingual parallel corpora, were constructed following nearly identical procedures.

1. Monolingual filtering was carried out in the following steps: 1) paragraphs were extracted from the CCNet corpus, 2) sentences were segmented, 3) duplicate sentences were removed, and 4) languages were identified using fastText (Grave et al., 2018) and langid.py (Lui and Baldwin, 2012).

2. Next, sentence alignment was carried out to build the parallel corpus. In CCMatrix, sentence embeddings were generated with the LASER model (Artetxe and Schwenk, 2019), and cross-lingual matches were identified using FAISS index search (Douze et al., 2025).

The NLLB corpus (NLLB Team et al., 2022) was constructed in a manner similar to CCMatrix. Monolingual filtering involved 1) language identification, 2) sentence segmentation, and 3) deduplication. Sentence alignment was then performed using scores derived from the LASER 3 model. In addition, heuristic filters based on sentence length and the proportion of symbols or numbers were applied.

While sentence alignment plays a key role in constructing parallel corpora, language identification is central to processing monolingual data. In this paper, we focus on monolingual filtering.

2.1.3 Parallel Corpus Filtering / Data Curation Task in WMT

The Conference on Machine Translation (WMT) organized shared tasks on parallel corpus filtering from 2018 to 2020 (Koehn et al., 2018, 2019, 2020). In these tasks, participants filtered noisy parallel corpora provided by the organizers and trained Transformer-based encoder-decoder models (Vaswani et al., 2023). Translation quality was then evaluated by the organizers. The target languages varied by year: in 2018, the focus was on a high-resource pair, German-English (De-En), while in 2019 and 2020, the focus shifted to low-resource pairs such as Nepali-English (Ne-En), Sinhala-English (Si-En), Pashto-English (Ps-En), and Khmer-English (Km-En).

The approaches adopted by participants in the 2020 shared task can be summarized as follows (Koehn et al., 2020): For monolingual processing, filtering involved 1) pre-filtering based on sentence length and character types, 2) language identification, and 3) language model scoring. For parallel processing, filtering relied on 4) LASER scores, 5) bidirectional cross-entropy between source and target languages, and 6) word-level translation scores.

In the WMT-2023 parallel data curation task (Sloto et al., 2023), Steingrimsson (2023) employed three language identification tools and discarded sentences where fewer than two out of three agreed.

To summarize, language identification and language model scores are central to monolingual fil-

Unit Type	Name	#Langs.	Description	Note
Sentence / Text	fastText[1]	176	Multiclass classifier based on skip-gram (Mikolov et al., 2013).	Grave et al. (2018); Joulin et al. (2016); Bojanowski et al. (2017)
	langid.py[2]	97	Naïve Bayes classifier based on a deterministic finite automaton	Lui and Baldwin (2012)
Token / Character	CMX	100	Multiclass classifier based on a feed-forward neural network; the language is identified at the token level	Zhang et al. (2018)
	CLD3[3]	107	Multiclass classifier based on a feed-forward neural network that averages character n-gram inputs	Google Chrome browser plugin.
	LanideNN	131	Bidirectional RNN classifier that identifies the language character by character from character embeddings	Kocmi and Bojar (2017)
	Equilid[4]	70	Three-layer neural network that identifies the language token by token	Jurgens et al. (2017)

Table 1: Summary of language identification.

tering. In practice, several language identifiers can be used in combination.

2.2 Language Identification

As noted in the previous section, language identification plays a central role in monolingual filtering.

Table 1 provides an overview of language identification methods. Almost all of the proposed approaches are learning-based and do not rely on manually crafted rules or dictionaries. These methods can be broadly categorized into two types: 1) sentence- or text-level identification and 2) token- or character-level identification.[5]

When sentence- or text-level language identifiers are used, code-mixed texts may be misclassified as the target language if the proportion of mixed content is small. In contrast, token-level identifiers determine the language at the token level, enabling filtering based on aggregated token-level results (Zhang et al., 2018). Since our proposed method adopts token-based identification, it is effective for handling code-mixed texts.

[1] https://fasttext.cc/docs/en/language-identification.html

[2] https://github.com/saffsd/langid.py

[3] https://github.com/google/cld3

[4] https://github.com/davidjurgens/equilid

[5] Unfortunately, we were unable to validate the performance of the token-level identifiers, as they were either difficult to obtain or no longer functional due to outdated implementations.

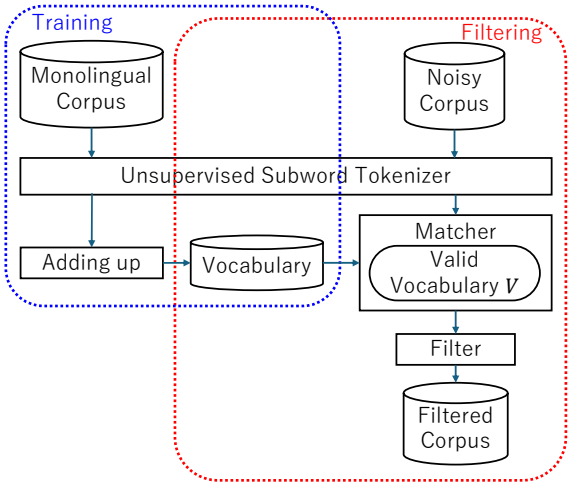

Figure 1: Structure of the proposed method.

3 Proposed Method

The proposed approach employs a binary classifier for each language. It automatically extracts vocabulary from a monolingual corpus, performs token-level identification by matching input tokens to the vocabulary, and applies filtering to remove invalid sentences. The overall architecture is illustrated in Figure 1. Training and filtering follow the steps outlined below.

Training

1. The monolingual corpus is tokenized into subwords. Although it is preferable for the corpus to contain only the target language, a certain

level of noise is acceptable, as discussed later. Therefore, web texts automatically collected using other language identifiers can be utilized.

For subword tokenization, we employ SentencePiece (Kudo and Richardson, 2018), an unsupervised tokenizer trained on the corpus.

2. Subword tokens are collected, sorted by frequency, and stored as a vocabulary.

Filtering

3. Load the vocabulary constructed in Step 2. To ensure coverage within the vocabulary limit VL, retain only the most frequent subwords and define them as the valid vocabulary V. Low-frequency subwords are excluded, as they typically represent noise.

4. The noisy corpus to be filtered is tokenized into subwords using the same tokenizer as in Step 1.

5. Match tokens against the valid vocabulary.

6. Sentences whose valid token ratio, the proportion of tokens contained in the valid vocabulary, is below the threshold TR are discarded as noise, while the remaining sentences are retained.

   Specifically, sentences satisfying the condition of the filter function $\mathrm{vocabFilter}(W)$ are output as the filtering result:

$$\begin{aligned} \mathrm{vocabFilter}(W) &= \frac{\sum_i \mathrm{match}(w_i)}{|W|} \\ &\geq TR, \\ \mathrm{match}(w) &= \begin{cases} 1 & \text{if } w \in V \\ 0 & \text{else} \end{cases}, \end{aligned}$$

   where W and w_i denote the sentence to be identified and the i-th token in W, respectively.

The proposed method has the following characteristics.

- In the overall framework, the proposed method effectively filters code-mixed texts by aggregating token-level identifications. This can be achieved by setting the threshold TR to a relatively high value.
- The proposed approach employs a language-specific binary classifier. In contrast to multi-class classifiers, which may require retraining on all languages when adding a new one, our method only constructs a binary classifier for the target language. This enables incremental addition of languages without reprocessing existing ones.
- Regarding the valid vocabulary, even if the monolingual corpus used for vocabulary acquisition contains noise, language identification remains feasible because function words dominate the top of the vocabulary hierarchy, content words occupy the middle, and noise tends to appear at the bottom (Table 2).
- The proposed method resembles filtering based on unigram language models. However, language models often assign low probabilities to content words, even in grammatically correct sentences. In contrast, our approach benefits from subword-level identification, allowing recognition of content words even when the full word is absent from the valid vocabulary.

Table 2 presents examples of valid vocabularies for German and Pashto obtained from the experiments described in the next section (English and Khmer vocabularies are provided in Appendix A). Although the total vocabulary size differs substantially across languages, two common patterns emerge: 1) symbols and function words, which constitute a large portion of each language, dominate the upper ranks; and 2) the lowest ranks consist almost entirely of noise, often including tokens from other languages. Consequently, removing the lower-ranked subwords yields the valid vocabulary for each language.

The vocabulary limit (VL) is defined as the threshold that separates valid subwords from invalid ones. In this study, VL was determined based on the cumulative coverage ratio, set to 99.5%. The resulting valid vocabulary sizes are 20,927 for English, 21,342 for German, 6,210 for Pashto, and 12,602 for Khmer. Subwords near this threshold often correspond to fragments of content words, making their classification inherently uncertain. To address this, we relax the token ratio threshold (TR) during the final filtering step, allowing for minor inconsistencies. In this work, TR is set to 0.9.

Order	Subword	Frequency	Cumulo-coverage
1	.	597M	3.71%
2	,	566M	7.23%
3	▁und	276M	8.95%
4	▁die	245M	10.47%
5	en	243M	11.98%
:			
21340	Wikipedia	13,937	99.49%
21341	teis	13,936	99.49%
21342	▁oxid	13,936	99.49%
21343	▁Malo	13,935	99.50%
21344	408	13,935	99.50%
21345	ARS	13,934	99.50%
:			
110927	Ấ	1	100.00%
110928	높	1	100.00%
110929	받	1	100.00%

(a) German vocabulary

Valid Vocabulary

Order	Subword	Frequency	Cumulo-coverage
1	د▁	6,789K	6.51%
2	په▁	3,282K	9.66%
3	او▁	2,198K	11.77%
4	.	2,131K	13.81%
5	کې▁	1,767K	15.50%
:			
6208	▁زهرا	266	99.49%
6209	کے	265	99.49%
6210	ئل	265	99.49%
6211	▁گرا	265	99.50%
6212	▁تنګ	265	99.50%
6213	▁فرشت	265	99.50%
:			
33473	▁	1	100.00%
33474	ټ	1	100.00%
33475	ã	1	100.00%

(b) Pashto vocabulary

Table 2: Examples of German and Pashto vocabularies obtained from the experiments in Section 4. Subwords contributing to a cumulative coverage of 99.5% were retained as the valid vocabulary.

4 Experiments

In this study, we refer to the proposed approach as 'vocabFilter'. Its effectiveness is evaluated based on machine translation quality using a filtered corpus.

4.1 Experimental Settings

4.1.1 vocabFilter Settings

We used CC-100 (Conneau et al., 2020) as the monolingual corpus for vocabulary acquisition.

We employed SentencePiece (Kudo and Richardson, 2018) as the tokenizer. The tokenizer model used in this study is the same as that adopted in the pretrained models mBART (Liu et al., 2020) and XLM-R (Conneau et al., 2020). This is the Unigram model of SentencePiece, which supports 100 languages and contains a vocabulary of approximately 250K subwords. Imamura and Utiyama (2024) reported that this model yields a low rate of unknown (UNK) tokens.

We set the hyperparameters to $VL = 0.995$ and $TR = 0.9$.

4.1.2 Parallel Corpus Filtering Task

We evaluated our approach following the parallel corpus filtering task of WMT. For this task, the organizers provided a noisy bilingual corpus together with sentence-level alignment scores (Table 3).

Lang. Pair	Noisy Corpus #Sents.	Noisy Corpus #Tokens	Selected Corpus #Sents.	Selected Corpus #Tokens
De-En	104M	1.0B	-	100M
Ps-En	1.02M	11M	-	5.0M
Km-En	4.17M	58M	-	5.0M

Table 3: Parallel corpora used in the experiment.

- In 2018, the target language pair was German-English, which is considered high-resource.
- In 2020, the targets were Pashto-English and Khmer-English, both regarded as low-resource pairs.
- We excluded the 2019 tasks from our evaluation because sentence alignment scores were not provided, and the objective of this study is monolingual corpus filtering.

The evaluation was carried out using the following procedure.

1. We filtered the noisy parallel corpus using the proposed method and comparative approaches.

2. After sorting the filtered results by alignment score, we selected the top sentences to reach a fixed number of tokens, counted on the English side. The thresholds were 100 million tokens for German and 5 million tokens for Pashto and Khmer (Table 3). Thus, we assumed that the amount of information in the parallel corpus remained constant regardless of the filtering method.

3. We trained the translation model using FairSeq (Ott et al., 2019), following the WMT shared task setup. Details of the hyperparameters are provided in Appendix B.

4. Finally, we evaluated translation quality on the test sets provided by WMT. For German, we used the devtest set, while for Pashto and Khmer, we combined the devtest and test sets. Translation quality was measured using sacreBLEU (Post, 2018) with the tokenizer for Flores-200 (NLLB Team et al., 2022; Goyal et al., 2022). BLEU was chosen because accurate surface translation was important in this study.

4.1.3 Comparative Methods

Following the baseline of the WMT-2020 shared task, we adopted fastText filtering as the baseline and combined it with the proposed method. Filtering was applied separately to the source and target languages.

4.2 Results

4.2.1 Translation Quality

Table 4 presents the BLEU scores obtained when each filter was applied to the source and target languages.

The effect of the proposed method, vocabFilter, varied across languages. For German (De $\leftrightarrow$ En), vocabFilter alone yielded lower translation quality than fastText alone; however, combining both methods improved the BLEU score over the baseline (i.e., fastText only).

Among the low-resource languages, Pashto (Ps $\leftrightarrow$ En) showed a smaller effect compared to German. However, translation quality improved when vocabFilter was applied to the source side in addition to fastText.

By contrast, for Khmer, both vocabFilter alone and its combination with fastText improved BLEU scores over the baseline, except when all filters were applied in the En $\rightarrow$ Km direction.

To summarize, translation quality tended to improve when vocabFilter was used in combination with fastText.

4.2.2 Number of Sentences Filtered out and Remaining

Table 5 presents the number of parallel sentences when language identification filtering was applied to both the source and target sides. 'Filtered' denotes the total number of sentences remaining after filtering, and 'Selected' denotes the number of sentences after selecting a fixed number of tokens.

First, focusing on 'Selected', we observe that although the number of English tokens is fixed, German tends to favor shorter sentences, as vocabFilter retained more sentences than fastText. In contrast, Pashto and Khmer show a slight decrease in sentence count, indicating a preference for longer sentences.

Next, focusing on 'Filtered', we can estimate the number of sentences for which fastText and vocabFilter produced different identification results. For example, in German, the difference between applying both filters and applying only fastText was 5.8M sentences (31.3M $-$ 25.5M), representing sentences accepted only by fastText. Similarly, sentences accepted only by vocabFilter totaled 20.6M (46.1M $-$ 25.5M), meaning that about 25% of the entire noisy corpus yielded different identification outcomes.

These differences suggest that fastText and vocabFilter evaluate sentences from different perspectives. Therefore, using them together can enhance language purity.

4.2.3 Example Sentences Filtered by vocabFilter

Table 6 provides example sentences accepted by fastText but rejected by vocabFilter. The sentences were tokenized using SentencePiece, and tokens shown in red indicate those that failed to match the valid vocabulary. Token ratio refers to the proportion of valid tokens; sentences with a token ratio below 0.9 were filtered out.

Regardless of language, these examples show that most code-mixed texts accepted by fastText were appropriately filtered out by vocabFilter. However, numeric tokens often caused identification failures (cf. No. 4, 7, and 8) because long numbers were frequently absent from the valid vocabulary. In addition, some tokens did not match the vocabulary even in sentences written in the cor-

fastText		vocabFilter		XX→En			En→XX		
Source	Target	Source	Target	De→En	Ps→En	Km→En	En→De	En→Ps	En→Km
✓	✓			29.5	8.8	7.3	27.5	10.7	14.8
		✓	✓	27.7 (-)	8.6	**8.1** (+)	25.4 (-)	10.6	**15.1** (+)
✓	✓	✓		**30.9** (+)	**9.1** (+)	**8.1** (+)	**28.6** (+)	**11.0** (+)	**15.1** (+)
✓	✓		✓	30.2 (+)	8.9	**8.1** (+)	28.1 (+)	**11.0** (+)	14.9
✓	✓	✓	✓	30.8 (+)	8.8	7.7 (+)	28.4 (+)	10.7	14.2 (-)

Table 4: BLEU scores when language identification was applied to each language. Bold indicates the highest score for each translation direction, and underlining denotes the second highest. The (+) and (-) symbols represent significant improvements or degradations, respectively, compared with fastText only (first row of data), based on bootstrap resampling with sacreBLEU ($p < 0.05$).

fastText	vocabFilter	De ↔ En			Ps ↔ En			Km ↔ En		
(both side)	(both side)	Noisy	Filtered	Selected	Noisy	Filtered	Selected	Noisy	Filtered	Selected
✓			31.3M	8.7M		560K	226K		2.27M	241K
	✓	104M	46.1M	12.3M	1.02M	593K	214K	4.17M	2.67M	221K
✓	✓		25.5M	7.64M		415K	214K		1.92M	215K

Table 5: Number of parallel sentences before and after filtering for each language identification.

Language	No.	Tokenized and matched example sentence	Token ratio
English	1	__" Ma hl zeit ", __cho reo graph er , __Theater __am __Wall , __War endorf	0.875
	2	__ -__ ي __رخصت __ملي __Em ba ssy __of __Afghanistan __in __Ott awa	0.750
	3	__ FOL LOW __US __ON __SOCIAL !	0.857
German	4	__Kontakt __B Berlin 1 __2015 -08- 26 T 17 :00 :26 +00:00	0.833
	5	__T sche chi en , __Li šov	0.857
	6	__Homepage __ | __Druck en __ | __Nach __oben	0.778
Pashto	7	__ 446 #__ 5__ __وړاندې __وړځي __مياشت و __7__by __Dari us s ssss	0.692
	8	5- رح __ __امريکا __شمالي __C ____ __د __اور پک FSX __& __P 3 D __2.5	0.867
	9	__Chang zhou __Daily s ____ __مح صو الت __يام لر ني __د Co . ، __Ltd	0.800
Khmer	10	__ផ្នែក __Put z meister __(__25 __)	0.857
	11	__English , __У к р а ї н с ь к а , __Français , __Español ... __សួម __មើល __ផ្នែក __ខ ន __បន្ថែម __ទៀត	0.800
	12	__Ma un fac turer __ ស្បែក ជើង បុរស សុវត្ថិភាព អ្នក	0.800

Table 6: Example sentences accepted by fastText but rejected by vocabFilter. Tokens shown in red indicate those that failed to match the valid vocabulary.

rect target language (cf. No. 3 and 5).

Conversely, even in languages that do not typically use Latin script, such as “Co. Ltd” in Pashto (No. 9) or “English” in Khmer (No. 11), frequently occurring words are recognized as valid tokens. Because vocabFilter does not rely solely on character-based decisions, high-frequency foreign words are not subject to filtering.

Although the proposed method could not perfectly filter all sentences, it offers a significant advantage by automatically removing code-mixed texts.

4.3 Influence of Hyperparameters

The proposed method involves two hyperparameters: 1) VL, the threshold for valid vocabulary, and 2) TR, the token ratio used to identify sentences. In this section, we examine how changes in these parameters affect translation quality.

Specifically, we measured translation quality by varying $VL \in \{0.95, 0.99, 0.995, 1.0\}$ and $TR \in \{0.8, 0.9, 0.95\}$. The results are presented in Table 7. This table reports the average BLEU scores across three language directions (De↔En, Ps↔En, and Km↔En) to illustrate overall trends. Detailed

TR \VL	0.95	0.99	0.995	1.0
0.8	**16.3**	15.9	15.8	15.5
0.9	15.1	<u>16.1</u>	15.8	15.5
0.95	12.5	<u>16.1</u>	16.0	15.5

(a) Average of XX → En.

TR \VL	0.95	0.99	0.995	1.0
0.8	**18.2**	**18.2**	18.0	17.8
0.9	17.4	18.1	18.1	17.6
0.95	12.5	17.9	17.9	17.8

(b) Average of En → XX.

Table 7: BLEU scores of vocabFilter under various settings of the hyperparameters VL and TR. Bold indicates the highest score, and underline indicates the second highest.

results for each language are provided in Appendix C.

Scores were noticeably lower for $VL = 1.0$ and for $TR \in \{0.9, 0.95\}$ when $VL = 0.95$. The remaining scores were similar, indicating that the proposed method performs well unless extreme hyperparameter settings are used.

5 Conclusion

In this paper, we proposed a simple monolingual filtering method. The method is a binary classifier that matches input tokens against an automatically acquired vocabulary. Because it operates at the subword level, it can handle unknown words relatively well. Adding a new language requires only tokenizing and aggregating a monolingual corpus, eliminating the need to retrain existing language models and enabling incremental language expansion.

We applied the proposed method to the filtering of noisy corpora and demonstrated improvements in machine translation quality. The method was particularly effective in removing code-mixed texts. When combined with other language identification techniques, it enables the creation of a cleaner and more consistent corpus.

We plan to release the program, along with the acquired vocabulary, after expanding the supported languages.

Limitations

Training the proposed method requires a monolingual corpus for the target language, although, as noted in Section 3, some degree of noise is acceptable.

The proposed method aims to improve corpus purity by removing code-mixed texts. However, this introduces a trade-off, as it reduces linguistic variety, particularly with respect to vocabulary. For example, user-generated content on social media often contains useful code-mixed expressions, yet the proposed method attempts to filter these out as well. In our experiments, we adopted BLEU as the evaluation metric, emphasizing surface-level similarity; however, it is also necessary to evaluate the method from additional perspectives, such as robustness to diverse inputs.

Ethics Considerations

Since the vocabulary used in this approach is automatically extracted from monolingual corpora, it may include problematic subwords if the corpora contain erroneous or inappropriate texts. Moreover, the filtering process cannot remove texts with problematic content.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Preprint*, arXiv:1607.04606.

Eleftheria Briakou, Colin Cherry, and George Foster. 2023. Searching for needles in a haystack: On the role of incidental bilingualism in PaLM's translation capability. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9432–9452, Toronto, Canada. 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.

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.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *Preprint*, arXiv:2401.08281.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *Preprint*, arXiv:1802.06893.

Kenji Imamura and Masao Utiyama. 2024. An empirical study of multilingual vocabulary for neural machine translation models. In *Proceedings of the Eleventh Workshop on Asian Translation (WAT 2024)*, pages 22–35, Miami, Florida, USA. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *Preprint*, arXiv:1607.01759.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 51–57, Vancouver, Canada. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2017. LanideNN: Multilingual language identification on character window. *Preprint*, arXiv:1701.03338.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on*

Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Steve Sloto, Brian Thompson, Huda Khayrallah, Tobias Domhan, Thamme Gowda, and Philipp Koehn. 2023. Findings of the WMT 2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 95–102, Singapore. Association for Computational Linguistics.

Steinthor Steingrimsson. 2023. A sentence alignment approach to document alignment and multi-faceted filtering for curating parallel sentence pairs from web-crawled data. In *Proceedings of the Eighth Conference on Machine Translation*, pages 366–374, Singapore. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *Preprint*, arXiv:2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337, Brussels, Belgium. Association for Computational Linguistics.

A Examples of Vocabulary in English and Khmer

Table 8 presents a subset of the valid vocabularies for English and Khmer obtained in the experimental setup described in Section 4. The vocabularies for German and Pashto are shown in Table 2.

B Hyperparameters of Machine Translator during Filtering Experiment

In the filtering experiments described in Section 4.1.2, we trained translation models using the hyperparameters listed in Table 9.

C Details of Influence of Hyperparameters

Table 7 shows the change in average translation quality across all languages when hyperparameters VL and TR are varied. Table 10 provides the corresponding language-specific details.

Order	Subword	Frequency	Cumulo-coverage
1	.	2,733M	3.60%
2	,	2,537M	6.94%
3	▁the	2,393M	10.08%
4	s	1,989M	12.70%
5	▁to	1,676M	14.91%
:			
20925	▁Pune	71,375	99.49%
20926	▁235	71,339	99.49%
20927	▁Edwin	71,338	99.49%
20928	bare	71,316	99.50%
20929	bana	71,316	99.50%
20930	▁Nou	71,307	99.50%
:			
169752	봤	1	100.00%
169753	雰	1	100.00%
169754	踴	1	100.00%

(a) English vocabulary

Valid Vocabulary

Order	Subword	Frequency	Cumulo-coverage
1	▁	13,362K	9.72%
2	។	1,013,K	10.46%
3	▁km▁	1,000,K	11.19%
4	▁ន៊	934K	11.87%
5	▁ា	931K	12.55%
:			
12600	tone	99	99.49%
12601	▁Plu	99	99.49%
12602	▁Aja	99	99.49%
12603	DV	99	99.50%
12604	Ali	99	99.50%
12605	18)	99	99.50%
:			
54417	Ư	1	100.00%
54418	ƙ	1	100.00%
54419	왔	1	100.00%

(b) Khmer vocabulary

Table 8: Examples of valid vocabularies for English and Khmer. Subwords covering up to 99.5% cumulatively were considered part of the valid vocabulary.

Model structure	
Architecture	Transformer
# of layers	5
Embedding dimension	512
FFN inner dimension	2,048
Attention heads	2
Other model settings	Share all embeddings Normalize before
Training	
Dropout	0.4
Attention dropout	0.2
ReLU dropout	0.2
Loss function	Label smoothed cross-entropy
Label smoothing	$\epsilon = 0.2$
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.98$)
Learning rate	1e-3
LR scheduler	Inverse square root
Warm-up steps	4,000
Global batch size	Roughly 16,000 tokens
Training epochs	100
Translation	
Beam width	5
Length penalty	1.2

Table 9: Hyperparameters for Model Training and Translation

	De → En				Ps → En				Km → En			
TR \VL	0.95	0.99	0.995	1.0	0.95	0.99	0.995	1.0	0.95	0.99	0.995	1.0
0.8	31.5	30.6	30.4	29.7	8.9	8.8	9.0	9.0	8.4	8.3	8.1	7.7
0.9	30.8	31.4	30.8	29.7	8.2	8.5	8.5	9.1	6.2	8.3	8.1	7.7
0.95	26.1	31.2	31.2	29.7	7.5	8.6	8.6	9.1	3.9	8.4	8.2	7.6

(a) XX → En directions.

	En → De				En → Ps				En → Km			
TR \VL	0.95	0.99	0.995	1.0	0.95	0.99	0.995	1.0	0.95	0.99	0.995	1.0
0.8	29.5	28.4	27.9	27.6	9.8	10.8	11.3	10.9	15.3	15.3	14.9	14.9
0.9	29.1	29.1	28.5	27.4	8.9	10.1	10.8	10.6	14.1	15.2	15.0	14.9
0.95	26.2	29.3	29.1	27.7	8.9	10.1	10.8	10.6	4.5	15.3	15.5	14.9

(b) En → XX directions.

Table 10: Performance of vocabFilter under different settings of hyperparameters VL and TR.