Neural Methods for Aligning Large-Scale Parallel Corpora from the Web for Southeast Asian Languages

Anonymous ACL submission

Abstract

We introduce neural methods to the hierarchical web mining approach of Paracrawl, showing large improvements. We apply these methods to web-scale parallel corpus mining for 9 South/Southeast/East Asian national languages, creating training resources for machine translation that yield better translation quality for most of these languages than existing publicly available datasets in OPUS, including CC-Matrix (Schwenk et al., 2021).

1 Introduction

The goal of this work is to apply neural methods to the task of parallel corpus mining from the web and to create large useful parallel corpora for languages that have not received much attention.

In addition to six Southeast Asian national languages (Burmese, Thai, Lao, Khmer, Vietnamese, Indonesian), we also included the South Asian languages Hindi and Nepali and the East Asian language Korean. These are mostly mid-resource languages, they have millions of speakers, mostly significant presence on the web, but have not received as much attention in the research community as European languages (Bañón et al., 2020), Indian languages (except, we include Hindi) (Siripragada et al., 2020), Chinese (Ziemski et al., 2016; Zhai et al., 2020), and Japanese (Morishita et al., 2022). Given that ACL is hosted this year in Southeast Asia, it is also timely that these languages receive some more attention.

Building on the work of the Paracrawl project (Bañón et al., 2020), we follow the same general sequence of steps: targeted web crawling, document alignment, sentence alignment, and parallel corpus filtering. Note that compared to the European-focused Paracrawl project, we have to deal with languages with less existing resources, mostly non-Latin script, and challenges such as lack of explicit word segmentation and even sentence boundary marking (the latter in the case of Thai).

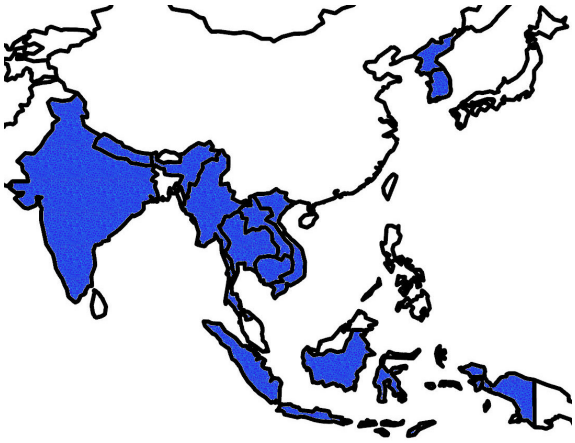

Figure 1: National languages covered: Hindi, Nepali, Burmese, Thai, Lao, Khmer, Vietnamese, Indonesian, Korean. We build parallel corpora for these languages paired with English.

In contrast to Paracrawl, we deploy neural methods in three steps: document alignment with an efficient Marian (Junczys-Dowmunt et al., 2018) neural machine translation model distilled from the multilingual NLLB (NLLB Team et al., 2022) model, sentence alignment with Vecalign (Thompson and Koehn, 2019), and using LASER for parallel corpus filtering (Chaudhary et al., 2019).

We obtain large parallel corpora of 1.5–8.3 million sentence pairs per language. We validate the usefulness of these corpora by showing better machine translation quality of up to +18.2 BLEU compared to CC-Matrix (Schwenk et al., 2021) for 7 languages and up to +13.0 BLEU compared to other existing parallel corpora on OPUS[1] (Tiedemann, 2009) for 6 languages (tied for another language). While this required significant computational resources, it was carried out using CPUs and consumer-grade GPUs (GTX 1080ti).

[1] https://opus.nlpl.eu/

2 Related Work

While the idea of mining the web for parallel data has been already pursued in the 20th century (Resnik, 1999), the most serious efforts have been limited to large companies such as Google (Uszkoreit et al., 2010) and Microsoft (Rarrick et al., 2011), or targeted efforts on specific domains such as the Canadian Hansards and Europarl (Koehn, 2005). More recently, large corpora have been released by broad web mining efforts, such as Paracrawl (Bañón et al., 2020) and CC-Matrix (Schwenk et al., 2021). A recent effort to assemble large-scale monolingual and parallel corpora is the EU Project High Performance Language Technologies (Aulamo et al., 2023).

Currently, there are two main approaches to extracting parallel sentence pairs from web documents: hierarchical and global mining. In hierarchical mining (as in Paracrawl), the task is broken up into the steps of identifying websites with parallel text, document alignment, sentence alignment, and sentence pair filtering. In global mining (as in CC-Matrix), all content is split up into sentences, each sentence represented by a cross-lingual sentence embedding and stored in one index per language. Then, sentences in one language are used to query the index of sentences in another language, using nearest neighbor search. There are also efforts that lie inbetween these two extremes, such as local mining in CC-Align (El-Kishky et al., 2020) where hierarchical mining is followed up to document alignment, and then sentences for each document are stored in an index and then queried regardless of the order of sentences in the document.

We follow the hierarchical mining approach. There has been varying amount of work on the steps in hierarchical mining. Matching documents pairs uses some similarity measure to compare the content of documents across languages. A common approach is to translate the non-English document into English and perform monolingual matching of words (Buck and Koehn, 2016) or n-grams (Dara and Lin, 2016; Uszkoreit et al., 2010). There have been some attempts to use document embeddings (Guo et al., 2019). Besides matching the URL (Le et al., 2016; El-Kishky et al., 2020) — e.g., `example.com/en/page.html` and `example.com/fr/page.html` — other structural information such the DOM-tree (Shi et al., 2006), links to the same images, links between pages, etc. have been used only rarely.

Sentence alignment has been a rich field of research dating back to the 1990s (Brown et al., 1991; Gale and Church, 1993). This also requires a similarity measure, defined over sentences or sequences of sentences. Typical features are sentence length and matches in a bilingual dictionary (Moore, 2002; Varga et al., 2005). Sennrich and Volk (2010) translate the non-English sentence and match the translation against the English sentence using the BLEU score. Vecalign (Thompson and Koehn, 2019) is a sentence alignment method that relies on bilingual sentence embeddings and achieves linear run time with a coarse-to-fine dynamic programming algorithm.

Finally, a lot of effort has been spend on developing methods for filtering noisy parallel corpora which are particularly harmful for neural models (Khayrallah and Koehn, 2018). Four shared tasks were dedicated to this problem (Koehn et al., 2018, 2019, 2020; Sloto et al., 2023). Besides basic simple filtering rules based on sentence or token length and their ratios (Kurfalı and Östling, 2019; Soares and Costa-jussà, 2019), typically a scoring function is used. Popular methods are based on the scores obtained by force-decoding the sentence pair with a machine translation model (Junczys-Dowmunt, 2018), and the cosine distance between cross-lingual sentence embeddings (Chaudhary et al., 2019). Mostly, classifiers that distinguish between genuine parallel sentence pair and misalignments have been trained, typically by using neural sentence representations (Açarçiçek et al., 2020; Esplà-Gomis et al., 2020; Xu et al., 2020; Tan et al., 2023).

3 Methods

3.1 Targeted Crawling

We follow the method of Paracrawl to crawl a list of targeted web sites. The list of websites to crawl was mainly obtained by using meta-data from CommonCrawl. Any website that has pages in English and any of the targeted languages and somewhat balanced ratio was selected and crawled with `httrack`[2], an open source web copying tool. We only follow links to web pages on the same web-domain. We stop crawling after crawling 50,000 pages for each website, both to avoid downloading duplicate webpages and due to computational limitations of subsequent processing steps.

[2] available at `https://www.httrack.com/`

Model	Vietnamese			Nepali			Thai		
	time	chrF	BLEU	time	chrF	BLEU	time	chrF	BLEU
MoE 54b official	-	62.3	43.8	-	66.9	48.1	-	57.8	36.9
Dense 3b official	-	61.5		-	65.9		-	56.8	
Dense 1b official	-	59.8		-	64.5		-	54.9	
Dense distilled 1b official	-	60.4		-	65.1		-	54.9	
Dense distilled 600m official	-	62.3		-	62.5		-	52.7	
Dense 3b QLoRA	207s	60.7	41.1	202s	62.2	41.1	238s	55.4	33.4
Dense distilled 1b QLoRA	61s	59.5	39.6	71s	63.2	42.2	74s	53.8	31.2
Dense distilled 1b	45s	59.8	39.2	44s	63.7	42.7	50s	54.1	31.5
Dense 1b	45s	58.9	38.6	44s	62.4	41.5	51s	54.2	31.6

Table 1: Speed/Quality trade-offs for different versions of NLLB, the model we distill. Translation time to translate the 1012 sentences of the Flores devtest set on a single GTX-1080 GPU (bottom). Official NLLB evaluations are in the top of the table. Based on these findings, we use the dense distilled 1 billion parameter model.

3.2 Distilling Machine Translation Models

Our document alignment approach requires the translation of all non-English web pages for a targeted language into English. Since this required the translation of a massive volume of text, we need an efficient but still sufficiently high-quality machine translation model.

The multilingual machine translation model NLLB (NLLB Team et al., 2022) covers 200 languages, including all the languages we target here. It comes in versions with 600 million to 54 billion parameters. However, even the smallest model would be computationally prohibitive given our technical means. Hence, we decided to distill these models into an efficient model that can be run on CPU via data distillation. In other words, we use the NLLB model to translate monolingual text and then use the resulting synthetic parallel corpus to train a faster model.

Table 1 shows machine translation quality and the time it takes to translate the 1012 sentences of the Flores-200 devtest set for three of our languages (Vietnamese, Nepali, and Thai) given different NLLB models. We explored the use of QLoRA, however we observed worse speed/quality trade-offs. We settled on using the dense distilled 1 billion parameter model. It gives reasonable performance at translation speeds of about 500 words per second on GPU.

The monolingual text for distillation is drawn from mC4[3] (Xue et al., 2021). We explored how much data we need to distill to get a reasonable Marian system. As illustrated in Figure 2, system quality plateaus at around 1 billion words of distilled data. Note that we exhausted the amount of monolingual data in mC4 for Lao, Khmer, and Burmese, so we distilled less data for these.

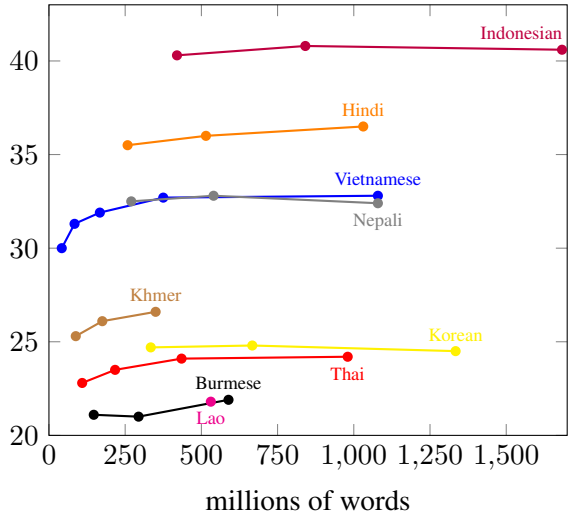

Figure 2: Amount of synthesized training data from the NLLB model and BLEU scores of distilled Marian models. For Lao, Khmer, and Burmese, we exhausted the monolingual data in mC4.

We generate synthetic parallel corpora by translating both from English and into English. The forward direction (X→English) is motivated by the idea of data distillation while backward translation (English→X) is well-established in the field since it has authentic text on the target side. As shown in Table 2, we find that forward translation gives better results, but combining both forward and backward translation fares generally best.

Note that we filter the synthesized corpus with LASER using a threshold of 1.05 (1.00 for Burmese and Lao, unfiltered for Hindi). See Section 3.5 more details on this method. We also added

[3] available at `https://huggingface.co/datasets/mc4`

Language	Forward	Backward	Both
Hindi	36.4	31.1	36.2
Nepali	30.8	30.1	33.6
Burmese	21.3	18.3	22.7
Thai	23.9	18.4	25.3
Lao	24.9	21.8	28.4
Khmer	25.5	13.3	26.3
Vietnamese	30.9	27.5	32.4
Indonesian	41.0	37.6	41.3
Korean	26.0	22.6	26.0

Table 2: BLEU scores for different data types for distillation: synthetic corpus generated by forward translation (X→English) or back translation (English→X). Forward translation fares better than backward translation, but combination of both is typically best.

all of OPUS to the training of Khmer, Vietnamese, and Korean distilled models.

The configuration of Marian (Junczys-Dowmunt et al., 2018) is given in Appendix A. The model is trained with guided alignment training and a vocabulary shortlist. The translation model uses quantized parameters for efficient vector integer computations support by Intel CPUs (8 bit, avx512). When translating web content, we observe translation speeds of about 1000 words per second in a single Intel Xeon Silver 4110 CPU core (contrast that to 500 words per second on a GPU for the NLLB model).

3.3 Document Alignment

Our document aligner follows the method by Buck and Koehn (2016). For a website where we found web pages in English and in the targeted language, we translate all the latter web pages into English and represent each document (i.e., web page) in form of word counts. Document similarity is measured by td/idf-weighted cosine distance between these representations. A greedy algorithm iteratively finds the best matching document pair and removes them from the pool of documents. The process terminates if documents in either language are exhausted.

The main difference to the Paracrawl approach is the use of a very efficient neural translation model instead of a statistical Moses model. The neural model has higher translation quality and is faster.

3.4 Sentence Alignment

We used Vecalign (Thompson and Koehn, 2019) as sentence aligner. It uses the cosine-distance between LASER embeddings with modified CSLS scoring (normalizing by distance to randomly chosen neighbors). But is also constrained by the order of the sentences in the pair of documents: Just like other sentence aligners (Hunalign, Bleualign, etc.), it may skip and merge sentences but it is not allowed to reorder them. Hence, it combines a powerful sentence matching method with the structural bias that documents are in almost all cases translated in sequence.

Documents were split into sentences with NLTK's sentence tokenizer. Thai required special treatment due to its lack of marking of sentence boundaries. We used the library `pythainlp` (Phatthiyaphaibun et al., 2023) for sentence splitting. We use LASER3, the latest version (Heffernan et al., 2022), that supports all our languages.

3.5 Parallel Corpus Filtering

The previous processing steps are geared towards high-recall instead of high-precision. In other words, we try to retain as much data as possible. This requires a final filtering step that removes noisy data, an open problem that has received much research attention.

We use LASER-based filtering (Chaudhary et al., 2019), using LASER3 (NLLB Team et al., 2022). This method embeds sentences in a cross-lingual embedding space, so that a English sentence and its translation should have identical representations. Hence, the distance between an English sentence embedding and a non-English sentence embedding is a measure for their meaning similarity. The exact formula to compute similarity between the two embedding vectors is the cosine distance, normalized by how similar each vector is to its closest neighbors in the embedding space.

We carried out limited experiments with the filtering threshold and chose a value of 1.00 for Nepali, Burmese, Thai, Lao, and Khmer and 1.05 for Hindi, Vietnamese, Indonesian, and Korean. We note that the more permissive threshold (1.00) worked better for the smaller corpora. For some languages we tried even lower threshold but that gave worse results.

4 Corpora

4.1 Corpus Statistics

We apply the processing pipeline to 9 languages. Table 3 gives detailed statistics. The pipeline succeeded to process between 5,854 (Burmese) and

Language	Crawls		Documents		Sentences		
	all	good	all	good	all	good	dedup
Hindi	13,605	6,603	4,033,751	80,856	52,919,986	5,989,651	4,823,444
Nepali	6,095	18,714	694,238	27,271	8,312,728	1,305,921	1,090,690
≥1.00		4,110		121,712		2,706,360	2,254,055
Burmese	5,854	2,166	790,360	13,662	9,769,167	890,742	718,709
≥1.00		3,437		69,236		2,002,212	1,674,072
Thai	14,012	2,266	3,349,364	11,680	61,466,936	1,470,556	1,190,997
≥1.00		6,751		50,533		2,761,013	2,218,153
Lao	4,177	2,441	492,723	13,186	7,513,409	1,158,534	936,986
≥1.00		3,401		63,820		2,391,972	2,004,028
Khmer	6,025	1,548	890,264	5,401	10,981,209	605,959	420,824
≥1.00		3,501		52,694		1,884,419	1,507,135
Vietnamese	32,765	9,927	6,951,765	124,453	80,256,711	8,735,317	6,473,708
Indonesian	20,031	8,067	5,443,448	116,734	77,507,912	10,304,822	7,260,778
Korean	24,500	11,069	7,758,116	136,847	128,828,741	11,014,387	8,298,299

Table 3: Detailed statistics on the crawled datasets, in terms of number of crawls of websites, number of aligned document pairs, and sentence pairs. The numbers below *good* specify counts for these categories that have valid sentence pairs after LASER filtering with threshold 1.05 (extra rows for languages where we applied the threshold 1.00) and deduplication. The deduplicated sentence pair count refers to a final global deduplication step.

Language	Ours	CC-Matrix	OPUS
Hindi	4.7m	15.1m	22.6m
Nepali	2.2m	19.6m	1.9m
Burmese	1.6m	10.0m	0.6m
Thai	1.8m	–	15.2m
Lao	1.9m	4.2m	4.2m
Khmer	1.3m	5.9m	0.6m
Vietnamese	6.4m	49.9m	18.8m
Indonesian	7.2m	56.8m	9.8m
Korean	8.2m	19.4m	19.7m

(a) Number of Segments

Language	Ours	CC-Matrix	OPUS
Hindi	75m	196m	296m
Nepali	32m	176m	12m
Burmese	28m	102m	8m
Thai	22m	–	152m
Lao	27m	40m	40m
Khmer	23m	66m	6m
Vietnamese	94m	780m	211m
Indonesian	110m	624m	88m
Korean	118m	205m	151m

(b) Number of English Words

Table 4: Size of parallel corpora, in millions, after length (≤80 words) and length ratio (≤9) filtering

32,765 (Vietnamese) website crawls. A small proportion (about 10%) of the crawls are repeat crawls, i.e., they crawled the same website again at a later time.

The next step is document alignment, resulting in 492,723 (Lao) to 7,758,116 (Korean) document pairs. Then comes sentence alignment, creating a raw corpus of 7,513,409 (Lao) to 128,828,741 (Korean) sentence pairs.

This corpus is filtered and deduplicated. We report how many good sentence pairs are retained when applying filtering to corpora from each crawl — which also includes deduplication: 605,959 (Khmer) to 11,014,387 (Korean). Then, deduplication is done again on the corpus combined across all crawl, reducing this number to 420,824 (Khmer) to 8,298,299 (Korean). These numbers are based on a filtering threshold of 1.05. For five of the languages we saw better results with a filtering threshold of 1.00, so we report these numbers as well. For Khmer, this retains 1,507,135 sentence pairs.

Working back from the filtered data, we can check how many document pairs had sentence pairs that survived filtering. For instance, this is the case for 136,847 of the 7,758,116 Korean–English document pairs. Applying the same calculation for web crawls, 11,069 of the 24,500 Korean web crawls yielded at least one sentence pair in the final filtered corpus.

4.2 Comparison to OPUS and CC-Matrix

We compare the size of the obtained corpora to existing data sets in Table 4. We combined all

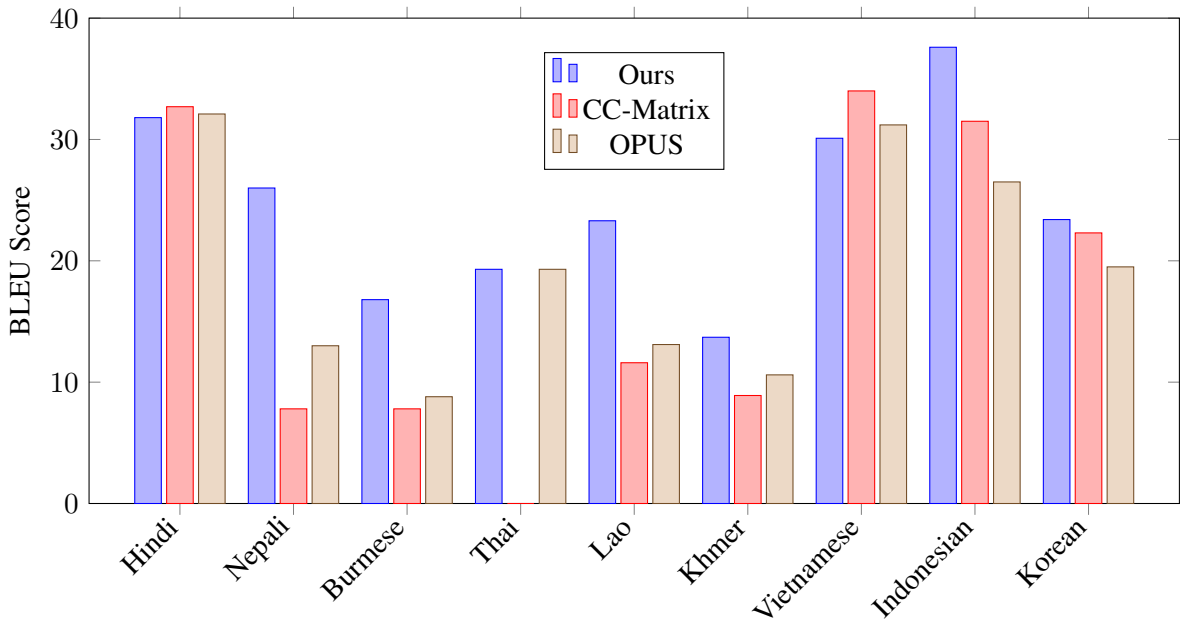

Figure 3: BLEU scores on neural machine translation systems build with our corpora, compared to existing corpora. We obtain better parallel corpora than anything previously existing for Nepali, Burmese, Lao, Khmer, Indonesian, and Korean, by a difference of +13.0, +8.0, +10.2, +3.1, +1.1 BLEU, respectively, compared to the better of CC-Matrix or OPUS. Our Thai corpus matches OPUS, For Hindi and Vietnamese, existing corpora are better. CC-Matrix does not include Thai.

parallel corpora available in OPUS, the popular platform for parallel data. We separated out CC-Matrix (which is also available on OPUS) since it is the method that is most similar to our approach and it is also typically the largest corpus on OPUS. CC-Matrix collected parallel sentences by matching sets of sentences from CommonCrawl solely based on the similarity of their LASER embeddings.

The table shows the number of segments and number of English words for each language. We count the number English words because it is a consistent measure across all languages and counting words for languages like Thai is problematic due to the lack of word spacing. The numbers are computed after another filtering step typically done for translation: we remove sentences longer than 80 words and sentence pairs where one sentence has more than 9 times as many words as the other.

Note that the sizes of the obtained corpora are smaller than CC-Matrix and only for Nepali, Burmese, and Khmer bigger than what already exists in OPUS (excluding CC-Matrix). We obtain a larger Indonesian corpus than what exists in OPUS in terms of number of words but not in number of segments. Our smallest corpus is Khmer–English (1.3 million segment pairs, 23 million words), the largest corpus is Korean–English (8.2 million segment pairs, 118 million words). Note that CC-Matrix does not include Thai.

5 Evaluation

Since our main motivation is to create parallel corpora for training machine translation systems, we evaluate them by using each to train a system, and measure each system's translation quality with spmBLEU (scarebleu `-tok flores200`) on Flores-200 (NLLB Team et al., 2022). We chose this test set since it exists for all our languages. It contains professional translations of English content drawn from Wikinews, Wikijunior, and Wikivoyage. We also computed scores with chrF++ which closely mirrors the spmBLEU results in terms of system ranking.

Machine translation systems were trained using Marian (Junczys-Dowmunt et al., 2018) using the setup as for our distilled translation models (see Section 3.2).

Results are shown in Figure 3. By our measure, we obtain better parallel corpora than anything previously existing for Nepali, Burmese, Lao, Khmer, Indonesian, and Korean, by a difference of +13.0, +8.0, +10.2, +3.1, +1.1 BLEU, respectively, compared to the better of CC-Matrix or OPUS. Our Thai–English corpus is as good as what is currently

Language	Ours	CC-M	OPUS	OPUS+CC-M	Ours+OPUS	Ours+OPUS+CC-M
Hindi	31.8	32.7	32.1	35.2	34.3	35.1
Nepali	26.0	7.8	13.0	21.5	25.2	25.8
Burmese	16.8	7.8	8.8	11.1	18.4	16.7
Thai	19.3	–	19.3	–	20.4	–
Lao	23.3	11.6	13.1	12.5	24.4	23.1
Khmer	13.7	8.9	10.6	17.0	19.8	21.3
Vietnamese	30.1	34.0	31.2	34.7	32.5	34.2
Indonesian	37.6	31.5	26.5	32.8	37.7	32.9
Korean	23.4	22.3	19.5	22.9	22.2	23.7

Table 5: Combining corpora: When combining our corpus with CC-Matrix and OPUS, we typically see improvements. The corpora are simply concatenated.

Language	Ours		Paracrawl	
	BLEU	Words	BLEU	Words
Nepali	26.0	32m	23.8	31m
Burmese	16.8	28m	11.5	13m
Khmer	13.7	23m	9.4	9m
Vietnamese	30.1	94m	31.1	123m
Korean	23.4	118m	21.8	88m

Table 6: Comparison of our neural methods with the statistical Paracrawl methods for document and sentence alignment.

in OPUS ($\pm$0). Only for Hindi and Vietnamese our data is fares worse (–0.9 and –3.9 BLEU, respectively).

It is worth noting that although our corpora are much smaller than CC-Matrix (by a factor of 2–8), we generally achieve better translation quality with them, indicating that the data is cleaner. These findings, however, allow only limited conclusion on the performance of the underlying methods (our hierarchical mining approach vs. the global mining approach of CC-Matrix) since they were executed on different, albeit quite similar, datasets (targeted crawling vs. pre-existing CommonCrawl).

6 Analysis

6.1 Combining Corpora

The three corpora we compare — OPUS, CC-Matrix, and ours — are obtained in quite different ways. Even though CC-Matrix and ours are both obtained by web crawling, CC-Matrix draws on CommonCrawl while we were doing our own crawling. Hence, we would expect that combining these corpora would lead to even better translation results.

Table 5 shows spmBLEU scores on Flores-200 devtest for the combinations OPUS+CC-Matrix, Ours+OPUS, and Ours+OPUS+CC-Matrix. For 3 languages (Khmer, Thai, and Korean) and almost Hindi–English, we do achieve the best results this way, while for Vietnamese the addition of our data slightly hurts (–0.5 BLEU) and for 3 languages (Burmese, Lao, Indonesian) the addition of the CC-Matrix corpus leads to worse results (–1.7, –1.3, and –4.8, respectively).

Note that we simply concatenated the corpora, and the typically larger size of the CC-Matrix corpus has bigger impact on the results. There are many other ways to combine and weigh corpora which should be explored in future work.

6.2 Comparison with Paracrawl Methods

Our pipeline makes two changes to the Paracrawl pipeline: use of a neural machine translation model for document alignment and sentence alignment based on neural sentence embeddings. Paracrawl uses a Moses-based statistical machine translation model and the lexicon-based Hunalign sentence aligner.

By running both the original pipeline and the pipeline with these changes, we can directly compare if the changes lead to an improved corpus. Results are shown in Table 6. We carried out this comparison only for 5 of the 9 languages due to the computation cost involved. Nevertheless, we covered both lower-resourced and higher-resourced languages. Except for Vietnamese (–1.0 BLEU), the neural methods lead to better results by a difference of +1.6 BLEU (Korean) to +5.3 BLEU (Burmese).

Since Vietnamese is an outlier here again (our new parallel corpus is also worse than CC-Matrix), we checked the execution of our pipeline for that language but could not find any obvious errors.

6.3 Computational Cost

We processed a total number of 127,064 web crawls. The size of the crawls has a very skewed distribution, with a few large crawls and a long tail of crawls that have only few web pages in the targeted languages. So, we can only make rough estimates about the processing cost.

Having said that, our document aligner takes about half an hour on average, of which half is spent on translation, summing up to about 2600 CPU days.

The sentence aligner takes about 6 minutes on average, the biggest computational cost being embedding of sentences with LASER, summing up to about 500 GPU days.

There is also signifcant time spent on extracting text from the web pages — we do not have reliable numbers on this. Note that this involves web crawls for which we ultimately do not find any content in targeted languages.

Sentence pair filtering takes tens of hours, training a neural model on a dataset a matter of a handful of days at most.

7 Limitations

The motivating goal for this work was to create high-quality parallel corpora for important languages that have previously not received much attention. The languages were also chosen due to their large difference to English, often even using non-Latin writing systems.

Given the vast computational cost involved, we only have limited results on the comparison of methods. For instance, a more fine-grained demonstration of the effectiveness of the document aligner and sentence aligner in isolation would be useful.

There are many more experiments that could be done with the data, such as more closely tracking how the quality of the machine translation model impacts the effectiveness of the document aligner. Another big area for follow-up research is how to best combine and filter different corpora for a language pair.

Finally, the only measure of translation quality that we offer is the translation quality of a machine translation system trained on a dataset. While this is ultimately what is most important for the consumer of this data, it also ignores many other aspects of data quality, such as toxic content or bias.

8 Risks

Our corpora may include harmful and violent. It may also contain content that is copyrighted. We claim that our use of web-crawled data follows fair-use exceptions but we will remove data if any specific requests are made, thus slightly altering the composition of the data.

9 Conclusions

We deployed neural methods to the Paracrawl processing pipeline, demonstrated their superiority against the previous statistical methods, and created high-quality parallel corpora for South/Southeast/East Asian languages that for 7 of the 9 languages lead to improvements in translation quality when building neural machine translation systems, for some languages dramatically.

We also spend significant effort on distilling NLLB models, reducing the computational cost by roughly doubling translation speeds, while using only a single CPU core vs. a full GPU.

We release all our corpora and models open source, with a liberal license for commercial and research use.

References

Haluk Açarçiçek, Talha Çolakoğlu, Pınar Ece Aktan Hatipoğlu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation*, pages 940–946, Online. Association for Computational Linguistics.

Mikko Aulamo, Nikolay Bogoychev, Shaoxiong Ji, Graeme Nail, Gema Ramírez-Sánchez, Jörg Tiedemann, Jelmer van der Linde, and Jaume Zaragoza. 2023. HPLT: High performance language technologies. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 517–518, Tampere, Finland. European Association for Machine Translation.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, ACL '91, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Buck and Philipp Koehn. 2016. Quick and reliable document alignment via TF/IDF-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678, Berlin, Germany. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Aswarth Abhilash Dara and Yiu-Chang Lin. 2016. Yoda system for wmt16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, Jaume Zaragoza-Bernabeu, and Felipe Sánchez-Martínez. 2020. Bicleaner at WMT 2020: Universitat d'alacant-prompsit's submission to the parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 952–958, Online. Association for Computational Linguistics.

William A Gale and Kenneth W Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Mandy Guo, Yinfei Yang, Keith Stevens, Daniel Cer, Heming Ge, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Hierarchical document encoder for parallel corpus mining. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 64–72, Florence, Italy. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

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.

Murathan Kurfalı and Robert Östling. 2019. Noisy parallel corpus filtering through projected word embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Thanh Le, Hoa Trong Vu, Jonathan Oberländer, and Ondřej Bojar. 2016. Using term position similarity and language modeling for bilingual document alignment. In *Proceedings of the First Conference on Machine Translation*, pages 710–716, Berlin, Germany. Association for Computational Linguistics.

Robert C Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France. European Language Resources Association.

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

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntorntip, and Can Udomcharoenchaikit. 2023. PyThaiNLP: Thai natural language processing in Python. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36, Singapore. Association for Computational Linguistics.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. MT detection in web-scraped parallel corpora. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 422–430. International Association for Machine Translation.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. 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.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 489–496. Association for Computational Linguistics.

Shashank Siripragada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A multilingual parallel corpora collection effort for Indian languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3743–3751, Marseille, France. European Language Resources Association.

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.

Felipe Soares and Marta R. Costa-jussà. 2019. Unsupervised corpus filtering and mining. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Weiting Tan, Kevin Heffernan, Holger Schwenk, and Philipp Koehn. 2023. Multilingual representation distillation with contrastive learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1477–1490, Dubrovnik, Croatia. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. 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 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann. 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, Beijing, China. Coling 2010 Organizing Committee.

Dániel Varga, Péter Halaácsy, András Kornai, Voktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005 Conference*, pages 590–596.

Runxin Xu, Zhuo Zhi, Jun Cao, Mingxuan Wang, and Lei Li. 2020. Volctrans parallel corpus filtering system for WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 985–990, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yuming Zhai, Lufei Liu, Xinyi Zhong, Gbariel Illouz, and Anne Vilnat. 2020. Building an English-Chinese parallel corpus annotated with sub-sentential translation techniques. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages

4024–4033, Marseille, France. European Language Resources Association.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Marian Configuration

The following configuration is used both for the distilled translation models that are used by the document aligner as well as for evaluating different corpora. We guided alignment training, with alignments generated by fast-align.

Model Configuration

```
dec-cell: ssru
dec-cell-base-depth: 2
dec-cell-high-depth: 1
dec-depth: 2
dim-emb: 256
enc-cell: gru
enc-cell-depth: 1
enc-depth: 6
enc-type: bidirectional
tied-embeddings-all: true
transformer-decoder-autoreg: rnn
transformer-dim-ffn: 1536
transformer-ffn-activation: relu
transformer-ffn-depth: 2
transformer-guided-alignment-layer: last
transformer-heads: 8
transformer-no-projection: false
transformer-postprocess: dan
transformer-postprocess-emb: d
transformer-preprocess: ""
transformer-tied-layers:
    []
transformer-train-position-embeddings:
false
type: transformer
```

Decoder Configuration

```
models
- model.intgemm.alphas.bin
shortlist:
- lex.s2t.gz
- false
beam-size: 1
normalize: 1.0
word-penalty: 0
mini-batch: 64
maxi-batch: 1000
maxi-batch-sort: src
workspace: 2000
max-length-factor: 2.5
gemm-precision: int8shiftAlphaAll
```

Training Parameters

```
–dim-vocabs 32000 32000
```

```
−max-length 200
−exponential-smoothing
−cost-type ce-mean-words
−mini-batch-fit -w 3000
−mini-batch 300
−maxi-batch 500
−sync-sgd −optimizer-delay 2
−learn-rate 0.0003 −lr-report
−lr-warmup 16000
−lr-decay-inv-sqrt 32000
−optimizer-params 0.9 0.98 1e-09
−clip-norm 0
−valid-freq 5000 −save-freq 5000
−disp-freq 1000
−valid-metrics bleu-detok ce-mean-words
−valid-mini-batch 64 −beam-size 1
−normalize 1
−early-stopping 100
```

Decoding Parameters

```
−beam-size 1 −mini-batch 32
−maxi-batch 100 −maxi-batch-sort src -w
128
−skip-cost −cpu-threads 1
```