CrossSum: Beyond English-Centric Cross-Lingual Summarization for 1,500+ Language Pairs

Anonymous ACL submission

Abstract

We present CrossSum, a large-scale cross-lingual summarization dataset comprising 1.68 million article-summary samples in 1,500+ language pairs. We create CrossSum by aligning identical articles written in different languages via cross-lingual retrieval from a multilingual summarization dataset and perform a controlled human evaluation to validate its quality. We propose a multistage data sampling algorithm to effectively train a cross-lingual summarization model capable of summarizing an article in any target language. We also introduce LaSE, an embedding-based metric for automatically evaluating model-generated summaries. LaSE is strongly correlated with ROUGE and, unlike ROUGE, can be reliably measured even in the absence of references in the target language. Performance on ROUGE and LaSE indicate that pretrained models fine-tuned on CrossSum consistently outperform baseline models. To the best of our knowledge, CrossSum is the largest cross-lingual summarization dataset and the first-ever that is not centered around English. We will release the dataset, alignment and training scripts, and the models to spur future research on cross-lingual summarization.

1 Introduction

Cross-lingual summarization (hereinafter XLS) is the task of generating a summary in a target language given a source text in another language. The task is challenging as it combines summarization and translation in one task, both challenging tasks in their own right. Earlier approaches to XLS thus employed pipeline methods such as translate-then-summarize (Leuski et al., 2003) and summarize-then-translate (Wan et al., 2010). Not only are they computationally expensive, having to use multiple models, but these approaches also suffer from error-propagation (Zhu et al., 2019) from one model to another, degrading the overall performance.

The success of sequence-to-sequence (seq2seq) models (Cho et al., 2014; Sutskever et al., 2014) and the advances in Transformer-based models (Vaswani et al., 2017) have aided in the emergence of end-to-end methods that can perform XLS with one single model (Zhu et al., 2019; Cao et al., 2020b). The availability of XLS datasets (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021) has also helped this task gain popularity in recent times. However, they cover only a few languages, contain a small number of samples for training and evaluation, or use English as the pivot language (i.e., the target language always remains English), thereby limiting their applicability to a great extent.

To democratize XLS beyond high-resource languages, in this work, we introduce CrossSum, a large-scale XLS dataset containing 1.68 million...
article-summary samples in 1,500+ language pairs. We align identical articles written in different languages via cross-lingual retrieval from the multilingual XL-Sum (Hasan et al., 2021) dataset. We perform a controlled human evaluation of CrossSum spanning nine languages from high-resource to low-resource and show that the alignments are highly accurate. We design a multistage sampling algorithm for successfully training models that can generate a summary in any target language for an article in any source language. For the first time, we perform XLS with CrossSum on a broad and diverse set of languages without relying on English as the standalone pivot language, consistently outperforming many-to-one and one-to-many models, as well as summarize-then-translate baselines.

We propose LaSE, an embedding-based metric for evaluating summaries when reference summaries may not be available in the target language but another language, potentially opening new doors for evaluating low-resource languages. Furthermore, we demonstrate the reliability of LaSE by its high correlation with ROUGE (Lin, 2004), the de-facto metric for summarization evaluation. We empirically set the similarity threshold as the 143. We use them to search identical contents across languages. For simplicity, we set two conditions for a summary $S_A$ in language $A$ to be matched with another summary $S_B$ in language $B$: 1. $S_B$ must be the nearest neighbor of $S_A$ among all summaries in $B$, and vice-versa. 2. The similarity between $S_A$ and $S_B$ must be above the threshold, $\tau$.

The similarity of a summary pair is measured by the inner product of their Language-agnostic BERT Sentence Representations (LaBSE) (Feng et al., 2022) (a unit vector for an input text sequence). We empirically set the similarity threshold as the average over all languages that maximized their respective $F_1$ score ($\tau = 0.7437$) in the BUCC mining tasks (Zweigenbaum et al., 2017).²

**Induced Pairs** We observed that many summary pairs, despite being nearest neighbors in their language pairs, were filtered out because of the threshold $\tau$. Although interestingly, both were matched with the same summary in a different language. Moreover, these pairs are prevalent if their languages are distant or low-resource. LaBSE uses contrastive learning (Guo et al., 2018; Yang et al., 2018).

The CrossSum Dataset

The most straightforward way of curating a high-quality XLS dataset is via crowd-sourcing (Nguyen and Daumé III, 2019). However, it may be difficult to find crowd workers having professional command over low-resource languages or distant language pairs. Moreover, scalability issues might arise due to the time and budget constraints for crowd-sourcing. Therefore, synthetic (Zhu et al., 2019) and automatic methods (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021) have gained traction over crowd-sourcing.

Automatic curation of an XLS dataset is simply to pair an article $A$ in a source language with the summary of an identical article $B$ written in a different target language (Figure 1), assuming the availability of a multilingual dataset having identical contents in different languages. Two contemporary works have compiled large-scale multilingual summarization datasets, namely XL-Sum (Hasan et al., 2021) (1.35M samples in 45 languages) and MassiveSumm (Varab and Schluter, 2021) (28.8M samples in 92 languages). Though substantially larger than the other, MassiveSumm is not publicly available. Since public availability is crucial for promoting open research, we opted for XL-Sum, distributed under a non-commercial license. Additionally, all articles of XL-Sum are crawled from a single source, BBC News. We observed that BBC publishes similar news content in different languages and follow similar summarization strategies. Hence adopting XL-Sum would increase the quality and quantity of the article-summary pairs.

Unlike previous automatic methods, there are no explicit links between identical articles in XL-Sum. Fortunately, language-agnostic sentence representations (Artetxe and Schwenk, 2019a; Feng et al., 2022) have achieved state-of-the-art results in cross-lingual text mining (Zweigenbaum et al., 2017; Artetxe and Schwenk, 2019b), and hence, we use them to search identical contents across languages. For simplicity, we set two conditions for a summary $S_A$ in language $A$ to be matched with another summary $S_B$ in language $B$: 1. $S_B$ must be the nearest neighbor of $S_A$ among all summaries in $B$, and vice-versa. 2. The similarity between $S_A$ and $S_B$ must be above the threshold, $\tau$.

The similarity of a summary pair is measured by the inner product of their Language-agnostic BERT Sentence Representations (LaBSE) (Feng et al., 2022) (a unit vector for an input text sequence). We empirically set the similarity threshold as the average over all languages that maximized their respective $F_1$ score ($\tau = 0.7437$) in the BUCC mining tasks (Zweigenbaum et al., 2017).²

**Induced Pairs** We observed that many summary pairs, despite being nearest neighbors in their language pairs, were filtered out because of the threshold $\tau$. Although interestingly, both were matched with the same summary in a different language. Moreover, these pairs are prevalent if their languages are distant or low-resource. LaBSE uses contrastive learning (Guo et al., 2018; Yang et al., 2018).

²The entire procedure is described in Appendix A.

²Around 90% $F_1$ is achieved using LaBSE in BUCC, hence not all CrossSum alignments will be correct. Therefore, in the following section, we further assess the quality of the alignments using human evaluation.
2019) to rank parallel sentences over non-parallel sentences. Since parallel pairs are mostly found for high-resource and linguistically close languages, we hypothesize that LaBSE fails to assign high similarity to sentences from languages that are not.

To include these pairs into CrossSum, we introduce the notion ‘induced pairs.’ Formally, two summaries $S_A, S_B$ in languages $A, B$ are induced pairs if they are nearest neighbors of each other in $A, B,$ their similarity score is below $\tau$, and both are matched with $S_C$ in language $C$ as valid pairs $(S_A, S_C), (S_C, S_B)$, or through a chain of valid pairs $(S_A, S_C), (S_C, S_D), \cdots, (S_Y, S_Z), (S_Z, S_B)$ in languages $\{C, D, \cdots, Y, Z\}$.

We thus incorporate the induced pairs into CrossSum through a simple graph-based algorithm. First, we represent all summaries as vertices in a graph and draw an edge between two vertices if the summaries are matched as valid pairs. Then we find the connected components in the graph and draw edges (i.e., induced pairs) between all vertices in a component. Again to ensure quality, before computing the induced pairs, we use the max-flow min-cut theorem (Dantzig and Fulkerson, 1955) considering the similarity scores as edge weights to limit the size of each component to 50 vertices (since ideally, a component should have at most 45 vertices, one summary from each language) and set their minimum acceptance threshold to $\tau' = (\tau - 0.10)$.

We finally assembled the original matched pairs and induced pairs to create the CrossSum dataset. Figure 6 (Appendix) shows the article-summary statistics for all language pairs in CrossSum. As evident from the figure, CrossSum is not centered only around the English language but rather distributed across multiple languages.

Implicit Leakage We initially made the train-dev-test splits respecting the original XL-Sum splits and performed an initial assessment of CrossSum by training a many-to-one model (articles written in any source language being summarized into one target language). Upon evaluation, we found very high ROUGE-2 scores (around 40) for many language pairs, even reaching as high as 60 for some (Figure 2). In contrast, Hasan et al. (2021) reported ROUGE-2 in the 10-20 range for the multilingual summarization task.

We inspected the model outputs and found that many summaries were precisely the same as the references. Through closer inspection, we found that all the articles, the summaries of which are exact copies of references, had their identical counterparts in some other language occurring in the training set. During training, the model was able to align the representations of identical articles (albeit written in different languages) and generate the same output by memorizing from the training sample. While models should undoubtedly be credited for being able to make these cross-lingual mappings, this is not ideal for benchmarking purposes as this creates unusually high ROUGE scores. We denote this phenomenon as ‘implicit leakage’ and make a new dataset split to avoid this. Before proceeding, we deduplicate the XL-Sum dataset using semantic similarity, considering two summaries $S_A, S'_A$ in language $A$ to be duplicates if their LaBSE representations have similarity above 0.95. We take advantage of the component graph mentioned previously to address the leakage and assign all article-summary pairs originating from a single component in the training (dev/test) set of CrossSum, creating an 80%-10%-10% split for all language pairs. Since identical articles no longer appear in the train set of one and the dev/test set

\[ 3 \text{XL-Sum has been deduplicated using lexical overlap methods only. But due to the risk of implicit leakage, which is not lexical, we further perform semantic deduplication.} \]
of another, the leakage is not observed anymore (Figure 2). We further validated this by inspecting the model outputs and found no exact copies.

3 Human Evaluation of CrossSum

To establish the validity of our automatic alignment pipeline, we conducted a human evaluation to study the quality of the cross-lingual alignments.

We selected all possible combinations of language pairs from a list of nine languages ranging from high-resource to low-resource to assess the alignment quality in different pair configurations (e.g., high-high, low-high, low-low) as per the language diversity categorization by Joshi et al. (2020). We chose three high-resource languages, English, Arabic, and (simplified) Chinese (category 4 and 5), three mid-resource languages, Indonesian, Bengali, and Urdu (category 3), and three low-resource languages, Punjabi, Swahili, and Pashto (category 1 and 2), as representative languages and randomly sampled fifty cross-lingual summary alignments from each language pair for annotation. As a direct evaluation of these pairs would require bilingually proficient annotators for both languages, which are practically intractable for distantly related languages (e.g., Bengali-Swahili), we resorted to a pivoting approach during annotation for language pairs that do not contain English. For a language pair \((l_1 - l_2)\), where \(l_1 \neq \text{en} \) and \(l_2 \neq \text{en}\), we sampled alignments \((x, y)\) such that \(\exists (x, e) \in (l_1 - \text{en})\) and \(\exists (y, e) \in (l_2 - \text{en})\), for an English article \(e\). In other words, we ensure that both the articles of the sampled cross-lingual pair have a corresponding cross-lingual pair with an English article. An alignment \((x, y)\) would be deemed correct if both \((x, e)\) and \((y, e)\) are correct. This formulation thus reduced the original problem to annotating samples from language pairs \((l_1 - \text{en})\) and \((l_2 - \text{en})\), where \(l_1\) and \(l_2\) are from the previously selected languages that are not English.

We hired bilingually proficient expert annotators adept in the language of interest and English. Two annotators labeled each language pair where one language is English. We presented them with corresponding summaries of the cross-lingual pairs (and optionally the articles themselves) and elicited yes/no answers to the question:

“Can the provided sequences be considered summaries for the same article?”

We deem a sequence pair accurate if both annotators judge it as valid. We show the accuracy of the language pairs in Figure 3.

As evident from the figure, the annotators judge the aligned summaries to be highly accurate, with an average accuracy of 95.67%. We used Cohen’s Kappa (Cohen, 1960) to establish the inter-annotator agreement and show the corresponding statistics in Table 2 in the Appendix.

4 Training & Evaluation Methodologies

In this section, we discuss the multistage sampling strategy for training cross-lingual text generation models and our proposed metric for evaluating model-generated summaries.

4.1 Multistage Language Sampling

From Figure 6, it can be observed that CrossSum is heavily imbalanced. Thus, training directly without upsampling low-resource languages may result in their degraded performance. Conneau et al. (2020) used probability smoothing for upsampling in multilingual pretraining and sampled all data points of a batch from one language. However, applying this technique to the language pairs in CrossSum would result in many batches having duplicate samples as many language pairs do not have enough examples. At the same time, many would not be sampled during training for lack of enough training steps (due also done to reduce annotation costs.)
Algorithm 1: Multistage sampling

**Input:** $D_{ij} \forall i, j \in \{1, 2, \ldots, n\}$: training data with tgt/src languages $L_i/L_j$; $c_{ij} \leftarrow |D_{ij}| \forall i, j \in \{1, 2, \ldots, n\}$; $m$: number of mini-batches.

1. Compute $q_i, q_{jj}$ using $c_{ij}$
2. **while** (Model Not Converged) **do**
3. 
4.   \[ batch \leftarrow \phi \]
5.   Sample $L_i \sim q_i$.
6.   for $k \leftarrow 1$ to $m$ **do**
7.     Sample $L_j \sim q_{jj}$.
8.     Create mini-batch $mb$ from $D_{ij}$
9.     \[ batch \leftarrow batch \cup \{mb\} \]
10. Update model parameters using $batch$.

4.2 Evaluating Summaries Across Languages

A sufficient number of reference samples are essential for the reliable evaluation of model-generated summaries. However, for many CrossSum language pairs, even the training sets are small, let alone the test sets (the median size is only 33). For instance, the Japanese-Bengali language pair only has 34 test samples, which is too few for reliable evaluation. But the in-language test samples for Japanese and Bengali are nearly 1k. Being able to evaluate against reference summaries written in the source language would thus alleviate this insufficiency problem by leveraging the in-language test set of the source language.

For this purpose, cross-lingual similarity metrics that do not rely on lexical overlap (i.e., unlike ROUGE) are required. Embedding-based similarity metrics (Zhang et al., 2020; Zhao et al., 2019) have recently gained popularity. We draw inspiration from them and design a similarity metric that can effectively measure similarity across languages in a language-independent manner. We consider three essential factors:

1. **Meaning Similarity**: The generated summary and the reference summary should convey the same meaning irrespective of their languages. Just like our alignment procedure from Section 2, we use LaBSE to compute the meaning similarity between the generated ($s_{gen}$) and reference summary ($s_{ref}$):

   \[
   MS(s_{gen}, s_{ref}) = \text{emb}(s_{gen}) \cdot \text{emb}(s_{ref})^T,
   \]

   where $\text{emb}(s)$ denotes the embedding vector output of LaBSE for input text $s$.

2. **Language Confidence**: The metric should identify, with high confidence, that the summary is indeed being generated in the target language. As such, we use the fastText language-ID classifier (Joulin et al., 2017) to obtain the language probability distribution of the generated summary and define the Language Confidence (LC) as:

   \[
   LC(s_{gen}, s_{ref}) = \begin{cases} 
   1, & \text{if } L_{ref} = \arg\max P(L_{gen}) \\
   P(L_{gen} = L_{ref}), & \text{otherwise}
   \end{cases}
   \]

3. **Length Penalty**: Generated summaries should not be unnecessarily long, and the metric should penalize long summaries. While model-based metrics may indicate how similar a generated summary is to its reference and language, it is unclear how they can be used to determine its brevity. As such, we adapt the BLEU (Papineni et al., 2002) brevity
penalty to measure the length penalty:

\[ LP(s_{gen}, s_{ref}) = \begin{cases} 1, & \text{if } |s_{gen}| \leq |s_{ref}| + c \\ \exp(1 - \frac{|s_{gen}|}{|s_{ref}| + c}), & \text{otherwise} \end{cases} \]

\( s_{gen} \) and \( s_{ref} \) may not be of the same language, and identical texts may vary in length across languages. Hence, we use a length offset \( c \) to avoid penalizing generated summaries slightly longer than the references. By examining the standard deviation of mean summary lengths of the languages, we set \( c = 6 \).

We finally define our metric, Language-agnostic Summary Evaluation (LaSE) score as follows.

\[ \text{LaSE}(s_{gen}, s_{ref}) = \text{MS}(s_{gen}, s_{ref}) \times \text{LC}(s_{gen}, s_{ref}) \times LP(s_{gen}, s_{ref}) \]

5 Experiments & Analysis

One model capable of generating summaries in any target language for an input article from any source language is highly desirable. However, it may not be the case that such a ‘many-to-many’ model (m2m in brief) would outperform many-to-one (m2o) or one-to-many (o2m) models\(^5\), which are widely-used practices for XLS (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021). In this section, we establish that the m2m model, in the presence of training samples from all possible language pairs, consistently outperforms m2o, o2m, and summarize-then-translate (s.+t.) baselines given equal training steps.

In addition to the proposed m2m model, we train five different m2o and o2m models using five highly spoken\(^6\) and typologically diverse pivot (i.e., the ‘one’ in m2o and o2m) languages: English, Chinese (simplified), Hindi, Arabic, and Russian. As another baseline, we use a summarize-then-translate pipeline. As fine-tuning pretrained language models (Devlin et al., 2019; Xue et al., 2021a) have shown state-of-the-art results on monolingual and multilingual text summarization (Rothe et al., 2020; Hasan et al., 2021), we fine-tune each model using a pretrained mT5 (Xue et al., 2021a) by providing explicit cross-lingual supervision\(^7\). We show the results on ROUGE-2 F1 and LaSE in Figures 4 and 5.

Results indicate that the m2m model consistently outperforms m2o, o2m, and s.+t., with an average ROUGE-2 (LaSE) score of 8.15 (57.15) over all languages tested, 3.12 (9.02) above s.+t. Moreover, compared to the o2m models on language pairs where the pivots are the targets, the m2m model scores 1.80 (5.84) over m2os, and on those where the pivots are the sources, 6.52 (51.80) over o2ms. We additionally perform a significance test of the m2m model’s performance in Appendix 3 and show it to be statistically superior to others.

Upon inspection, we found the m2o models to be able to generate non-trivial summaries, while the o2m models completely failed to produce cross-lingual summaries, performing in-language summarization for all targets\(^9\). s.+t. performed well on high-resource languages but poorly on low-resource ones. Inspection revealed this to be a limitation of the translation model used in the pipeline.

<table>
<thead>
<tr>
<th>Target Lang.</th>
<th>ROUGE-2 vs.</th>
<th>LaSE-in-lang vs. LaSE-out-lang.</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>0.976/0.939</td>
<td>0.993/1.000</td>
</tr>
<tr>
<td>Arabic</td>
<td>0.903/0.987</td>
<td>0.968/0.942</td>
</tr>
<tr>
<td>Chinese</td>
<td>0.983/1.000</td>
<td>0.996/1.000</td>
</tr>
<tr>
<td>Indonesian</td>
<td>0.992/0.975</td>
<td>0.872/0.828</td>
</tr>
<tr>
<td>Bengali</td>
<td>0.947/0.902</td>
<td>0.819/0.771</td>
</tr>
<tr>
<td>Urdu</td>
<td>0.997/0.951</td>
<td>0.774/0.828</td>
</tr>
<tr>
<td>Punjabi</td>
<td>0.988/0.963</td>
<td>0.881/0.885</td>
</tr>
<tr>
<td>Swahili</td>
<td>0.990/0.951</td>
<td>0.979/0.885</td>
</tr>
<tr>
<td>Pashto</td>
<td>0.994/0.987</td>
<td>0.883/0.885</td>
</tr>
</tbody>
</table>

Table 1: Correlation analysis of ROUGE-2 and LaSE.

How reliable is LaSE? At first, we validated the reliability of LaSE by showing its correlation with ROUGE-2. We took different checkpoints of the in-language summarization model used in s.+t. and computed ROUGE-2 and LaSE for the nine languages in Section 3 for each checkpoint. The correlation coefficients of the computed scores are shown in the second column of Table 1. For all languages (from high- to low-resource), LaSE has a near-perfect correlation with ROUGE-2.

However, the purpose of LaSE is to show that it is language-agnostic and can even be computed in the absence of references in the target language.

\(^5\)Discussed in detail in Appendix D.

\(^6\)https://w.wiki/Pss

\(^7\)Zero-shot cross-lingual transfer discussed in Appendix E.3.

\(^8\)A detailed description of the training procedures and hyperparameter choices are detailed in Appendix E.1.

\(^9\)We hypothesize that varying the target language in a batch hampers the decoder’s ability to generate from a specific language, possibly because of the vast diversity of target languages in the batch (discussed further in Appendix F).
Therefore, we evaluate the summaries with references in a different language from the target using the m2m model. For each target language, we first compute the standard LaSE for different source languages (denoted as LaSE-in-lang). We again compute LaSE after swapping the reference texts with the references in the language of the input text\(^\text{10}\) (denoted as LaSE-out-lang). We then show the correlation between the two variants of LaSE in the third column of Table 1\(^\text{11}\) for each target language. Results show a substantial correlation between the two variants of LaSE for all languages.

\(^{10}\)Our curation method ensures that such summaries always exist in the corresponding test sets.

\(^{11}\)Since many test sets of the language pairs from Section 3 have too few samples for reliable evaluation (e.g., Punjabi-Pashto), for each target language, we use only the top-5 source languages by the number of their test set samples.

From these two experiments, we can conclude that LaSE is ideal for summary evaluation and can be computed in a language-independent manner.

6 Related Works

Pipeline-based methods were popular at the beginning stages of XLS research (Leuski et al., 2003; Orasan and Chiorean, 2008; Wan et al., 2010), breaking it into two sequential summarization and translation tasks. End-to-end methods that performed XLS with a single model gained popularity with the emergence of neural models. Ayana et al. (2018) used knowledge distillation (Hinton et al., 2015) to train a student XLS model from two summarization and translation teacher models. Using a synthetic dataset, Zhu et al. (2019); Cao et al. (2020a) performed XLS with a dual Transformer
(Vaswani et al., 2017) architecture in a multitask framework, while Bai et al. (2021) proposed a single encoder-decoder for better transfer across tasks. Until recently, XLS was limited to English-Chinese only due to the lack of benchmark datasets. To promote the task beyond, Ladhak et al. (2020) introduced Wikilingua, a large-scale many-to-one dataset with English as the pivot language, while Perez-Beltrachini and Lapata (2021) introduced XWikis, containing 4 languages in 12 directions.

7 Conclusion & Future Works

In this paper, we presented CrossSum, a large-scale, non-English-centric XLS dataset containing 1.68 million samples across 1500+ language pairs. CrossSum provides the first publicly available XLS dataset for many of these pairs. Performing a limited-scale human evaluation of CrossSum, we introduced a multistage sampling algorithm for general-purpose cross-lingual generation and a language-agnostic metric for evaluating summaries when references in the target languages may not be available. Additionally, we demonstrated that training one multilingual model can help towards better XLS than baselines. We also shed some light on the potential to perform zero-/few-shot XLS with CrossSum.

In the future, we will investigate the use of CrossSum for other summarization tasks, e.g., multi-document (Fabbri et al., 2019) and multi-modal summarization (Zhu et al., 2018). We would also like to explore better techniques for m2m, zero-shot, and few-shot summarization.
Limitations

Though we believe that our work has many merits, some of its limitations must be acknowledged. Despite exhaustive human annotation being the most reliable means of ensuring the maximum quality of a dataset, we had to resort to automatic curation of CrossSum due to the enormous scale of the dataset. As identified in the human evaluation, not all of the alignments made by LaBSE are correct. They are primarily summaries describing similar (i.e., having a substantial degree of syntactic or semantic similarity) but non-identical events. LaBSE also fails to penalize numerical mismatches, especially if the summaries depict the same event.

Consequently, any mistake made by LaBSE in the curation phase may propagate to the models trained using CrossSum. And since LaBSE is a component of the proposed LaSE metric, these biases may remain unidentified by LaSE in the evaluation stage. However, no matter which automatic method we use, there will be such frailties in these extreme cases. Since the objective of this paper is to not scrutinize the pitfalls of LaBSE but rather to use it as a means of curation and evaluation, we deem LaBSE the best choice due to its extensive language coverage and empirical performance in cross-lingual mining among existing alternatives.

Ethical Considerations

License CrossSum is a derivative of the XL-Sum dataset. XL-Sum has been released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), allowing modifications and distributions for non-commercial research purposes. We are adhering to the terms of the license and will also release CrossSum under the same license.

Generated Text All of our models use the mT5 model as the backbone, which is pretrained on a large multilingual text corpus. For a text generation model, even small amounts of offensive or harmful texts in pretraining could lead to dangerous biases in generated text (Luccioni and Viviano, 2021). Therefore, our models can potentially generate offensive or biased content learned during the pretraining phase, which is beyond our control. Text summarization systems have also been shown to generate unfaithful and factually incorrect (albeit fluent) (Maynez et al., 2020) texts. Thus, we suggest carefully examining the potential biases before considering them in any real-world deployment.

Human Evaluation Annotators were hired from the graduates of an institute that provides professional language training for many languages, including the ones evaluated in Section 3. Each annotator was given around 200-250 sequence pairs to evaluate. Each annotation took around one and a half minutes on average, with a total of approximately 5-6 hours for annotating the whole set. Annotators were paid hourly as per the standard remuneration of bilingual professionals in local currency.

Environmental Impact About 25 models were trained in total in this paper. Each model was trained for about 3 days on a 4-GPU Tesla P100 server. Assuming 0.08 kg/kWh carbon emission, less than 175kg of carbon was released into the environment in this work, which is orders of magnitude below the most computationally demanding models.

References


Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim


Appendix

A Aligning Summaries using LaBSE

In Section 2, we curate CrossSum by aligning identical summaries in different languages. It might be argued why the articles themselves were not used for the alignment process. Initially, we experimented with whole-article embeddings. However, this resulted in many false-negative alignments, where similarity scores between identical articles across languages were relatively low (verified manually between English and the authors’ native languages). This is most likely attributed to the 512-token limit of LaBSE and different sequence lengths of those articles due to different languages having different subword segmentation fertility (Ács, 2019). This would entail that identical articles in different languages might be truncated at different locations, resulting in discrepancies between their embeddings. As observed in the BUCC evaluation, LaBSE is well-suited for sentence-level retrieval. Since summaries are good representatives of entire articles, we finally chose summaries as our candidates for the alignment.

B Inter-annotator Agreement of Human Evaluation

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>Cohen’s Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic-English</td>
<td>0.82</td>
</tr>
<tr>
<td>Chinese-English</td>
<td>0.73</td>
</tr>
<tr>
<td>Indonesian-English</td>
<td>0.73</td>
</tr>
<tr>
<td>Bengali-English</td>
<td>0.73</td>
</tr>
<tr>
<td>Urdu-English</td>
<td>0.76</td>
</tr>
<tr>
<td>Punjabi-English</td>
<td>0.71</td>
</tr>
<tr>
<td>Swahili-English</td>
<td>0.78</td>
</tr>
<tr>
<td>Pashto-English</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 2: Language pair-wise kappa scores.

C Statistical Significance

While the scores obtained from the experiments in Section 5 are a telling sign that the proposed m2m model performs better than the others, the differences are very close in many language pairs. Therefore, a statistical significance test is still warranted to support our claim further. As such, for each language pair experimented on, we performed the Bootstrap resampling test (Koehn, 2004) with the m2m model against the best performing model among the others in a one vs. all manner: if m2m has the best score (ROUGE-2/LaSE), we compare it with the model with the second-best score, and if m2m is not the best, we compare it with the best. Results ($p < 0.05$) reveal that in more than 42% language pairs tested, m2m is significantly better, and in less than 10% pairs, it is considerably worse. Details presented in Table 3.

| Pivot Metric Better Worse Insignificant |
|---------------------------------------|-------------------------------|
| x-en R-2/LaSE 8/18 2/2 25/15          |
| en-x R-2/LaSE 20/15 3/14 12/6         |
| x-zh R-2/LaSE 11/13 0/0 23/21         |
| zh-x R-2/LaSE 17/12 1/2 16/20         |
| x-hi R-2/LaSE 18/15 1/6 15/13         |
| hi-x R-2/LaSE 19/15 0/6 15/13         |
| x-ar R-2/LaSE 6/15 2/3 26/16          |
| ar-x R-2/LaSE 23/15 1/5 10/14         |
| x-ru R-2/LaSE 6/11 2/7 26/16          |
| ru-x R-2/LaSE 19/13 2/7 13/14         |

Table 3: Significance test on different pivot languages.

D Modeling Details

D.1 Choice of Pretrained Model

Many pretrained multilingual text-to-text models are currently available, e.g., mBART (Liu et al., 2020), CRISS (Tran et al., 2020), MARGE (Lewis et al., 2020), and mT5 (Xue et al., 2021b). While mBART and mT5 are pretrained with multilingual objectives, CRISS and MARGE are pretrained with a cross-lingual one, which better suits our use case. However, we choose mT5 for fine-tuning because of its broad coverage of 101 languages with support for 41 of the 45 languages from CrossSum, in contrast to only 15 languages in mBART or CRISS and 26 in MARGE.

D.2 Summarize-then-translate (s. + t.)

The primary reason for using summarize-then-translate rather than translate-then-summarize is the computational cost between these two. Available translation models only work for short sequences and are unsuitable for long documents. One solution is to segment the documents into sentences and then translate them. But that increases the compute overhead, and translations suffer from loss of context. We use a multilingual summarization model (Hasan et al., 2021) coupled with the multilingual machine translation model, M2M-100 (Fan et al., 2021), for our pipeline.
Figure 6: A bubble plot depicting the article-summary frequencies of CrossSum. The radii of the bubbles are proportional to the number of samples for the corresponding language pair (exact numbers are in Table 4). Languages are ordered by the language taxonomy from Joshi et al. (2020). To show better contrast between language pairs, we color a bubble cyan if its frequency is below 500 (1218 pairs), red for 500 to 5000 (688 pairs), and blue for frequencies exceeding 5000 (52 pairs).

D.2.1 Multilingual Summarization

The pipeline first performs in-language summarization (the language of the summary is the same as that of its input article) and then translates the summary into the desired target language. We train our own model for summarization as the model released by Hasan et al. (2021) has been rendered unusable due to the change in the dataset split. We train our component graphs to curate the in-language dataset splits. We consider articles having no identical counterpart in any other language as single node components in the component graph. As before, we assign all articles originating from a single component to the training (dev/test) set of the dataset, extending them to the in-language splits too. We then train the multilingual model by fine-tuning mT5 with the in-language splits, sampling each batch of 256 samples from a single language with a sampling factor of $\alpha = 0.5$.

D.2.2 Multilingual Translation

For multilingual translation, we used M2M-100 (Fan et al., 2021) (418M parameters variant), a
Figure 7: Training on the dataset respecting the original XL-Sum splits causes absurdly high ROUGE scores (marked red) in many-to-one models due to implicit data leakage. Therefore, we split taking the issue into account and consequently, models trained on the new set (marked blue) do not exhibit any unusual spike in ROUGE-2.

**D.3 Many-to-One (m2o) Model**

Many-to-one training is standard for evaluating cross-lingual summarization. In these models, the language of the source text can vary, but the target language remains the same, i.e., as the pivot language. Instead of sampling all samples of a batch from the same language pair, we sample 8 mini-batches of 32 samples using a sampling factor of $\alpha = 0.25$, the source side of each originating from a single language while the target language remains fixed. We then merge the mini-batches into a single batch and update the model parameters. This is to ensure that there are not many duplicates in a single batch (if all 256 samples of a batch are sampled from a single language pair, there might be many duplicates as many language pairs do not have 256 training samples) and the model still benefits the advantages of low-resource upsampling.

**D.4 One-to-many (o2m) Model**

o2m models are complementary to m2o models: we train them by keeping the source language fixed and varying the target language. We upsample the low-resource target languages with the same sampling factor of $\alpha = 0.25$ and merge 8 mini-batches of 32 samples each, analogous to m2o models.

**D.5 Many-to-many (m2m) Multistage Model**

This is the model obtained from the Algorithm 1. In contrast to standard language sampling (Conneau et al., 2020), we sample the target language and...
then choose the source based on that decision. We use batch size 256, 8 mini-batches with size 32 and \( \alpha = 0.5, \beta = 0.75 \).

**D.6 Many-to-many (m2m) Unistage Model**

This algorithm is similar to standard language sampling, the difference being that languages are sampled as pairs from all possible combinations. Instead of sampling one language pair at each training step, we sample 8 pairs, one for each mini-batch of size 32. We then merge the mini-batches into a single batch of 256 samples before updating the model parameters. We use a sampling factor of \( \alpha = 0.25 \).

In all models, we discarded a language pair from training if it had fewer than 30 training samples to prevent too many duplicates in a mini-batch. The training was done together with the in-language samples.

**E Experimental Details**

**E.1 Training Setups**

Fine-tuning generation models is compute-intensive, and due to computational limitations, we fine-tune all pretrained models for 25k steps with an effective batch size of 256, which roughly takes about three days on a 4-GPU NVIDIA P100 server. We use the base variant of mT5, having 250k vocabulary, 768 embedding and dimension size, 12 attention heads, and 2048 FFN size, with 580M parameters. We limit the input to 512 and output to 84 tokens. All models are trained on the respective subsets of the CrossSum training set.

**E.2 Inference**

During inference, we jump-start the decoder with language-specific \texttt{BOS} (beginning of sequence) tokens (Johnson et al., 2017) at the first decoding step for guiding the decoder to generate summaries in the intended target language. We use beam search (Medress et al., 1977) with the beam size 4 and use a length penalty (Wu et al., 2016) of 0.6. We limit ourselves only to the languages supported by mT5, fastText, and M2M-100.

**E.3 Zero-shot Cross-lingual Transfer**

The previous experiments were done in a fully supervised fashion. However, for many low-resource language pairs, samples are not abundantly available. Hence, it is attractive to be able to perform zero-shot cross-lingual generation (Duan et al., 2019) without relying on any labeled examples.

To this end, we fine-tuned mT5 with the in-language (both source and target are in the same language) samples only in a multilingual fashion and, during inference, varied the target language. Unfortunately, the model totally fails at generating cross-lingual summaries and performs in-language summarization instead.

We also fine-tuned m2o models in a zero-shot setting (with only the in-language samples of the target language) in a monolingual fashion. Here, the models are able to generate non-trivial summaries for some language pairs but still lag behind fully supervised models by a significant margin (Figure 10 and 11).

Furthermore, we ran inference with the m2m model on distant low-resource language pairs that were absent during training. Their LaSE scores were substantially below supervised pairs, meaning zero-shot transfer in supervised multilingual models (Johnson et al., 2017) shows weak performance as well.

We do not perform any few-shot experiments and leave them as potential future directions.

**F Ablation Studies**

We make several design choices in the multistage sampling algorithm. We break them into two main decisions:

1. Making mini-batches and sampling the language pair for each mini-batch.
2. Keeping either the source or the target language fixed for each batch.

To verify that these choices indeed affect performance positively, we train five different models for ablation:

1. Sampling the language pair in mini-batches in one stage only and then merging them into large batches before updating model parameters: m2m-unistage.
2. Sampling the language pair with large batches of 256 samples without mini-batching: m2m-large.
3. Multistage sampling keeping only the target language fixed in a batch: m2m-tgt \([our\ proposed\ model]\).
4. Multistage sampling keeping only the source language fixed in a batch: m2m-src; i.e., the complement of our proposed model.

5. Multistage sampling keeping either the source or the target language fixed (with equal probability) for each batch: m2m-src-tgt.

We benchmark on all the language pairs done previously and show the mean ROUGE-2 and LaSE scores in Table 5.

<table>
<thead>
<tr>
<th>Model</th>
<th>Scores</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROUGE-2</td>
<td>Better</td>
</tr>
<tr>
<td>m2m-large</td>
<td>8.31/57.45</td>
<td>122</td>
</tr>
<tr>
<td>m2m-unistage</td>
<td>7.51/55.36</td>
<td>191</td>
</tr>
<tr>
<td>m2m-tgt</td>
<td>8.15/57.15</td>
<td>289</td>
</tr>
<tr>
<td>m2m-src</td>
<td>4.44/26.75</td>
<td>34</td>
</tr>
<tr>
<td>m2m-src-tgt</td>
<td>6.47/42.55</td>
<td>89</td>
</tr>
</tbody>
</table>

Table 5: ROUGE-2 and LaSE scores for ablation.

As can be seen from the table, m2m-large, the standard m2m model, has the best average ROUGE-2/LaSE scores among all m2m variants. This begs the question of whether our proposed multistage sampling is, after all, needed or not. But the scores of the proposed m2m-tgt model does not fall much below. Therefore, we show statistical significance test results of all m2m models, comparing them against m2o, o2m, and s.+t. in one vs. all manner.

Significance results paint a different picture: m2m-tgt triumphs over all other models, getting significantly better results on 42% language pairs, more than double the m2m-large model. We inspected the results individually and found that the results are notably better on language pairs that are not adequately represented in the training set. m2m-tgt performs comparatively worse on high-resource language pairs, which we believe is a fair compromise to uplift low-resource ones. As m2m-large can sample a pair only once per batch, it fails to incorporate many language pairs due to them having insufficient participation during training. On the other hand, our proposed multistage sampling algorithm performs well in this regard by sampling in two stages.

While m2m-tgt outperforms all the rest, m2m-src falls behind all other models by a large margin. This phenomenon also has the same trend as the results in Section 5, where o2m models failed at generating cross-lingual summaries. This is in line with our hypothesis made in Footnote 9, as m2m-src and m2m-tgt mimic the training settings of the o2m and m2o models, respectively, at the batch level. The m2m-src-tgt is the middle ground between m2m-src and m2m-tgt and, likewise, scores between these two. In our opinion, the performance dynamics between the m2o (m2m-tgt) and o2m (m2m-src) models is an interesting finding and should be studied in depth as a new research direction in future works.
Figure 8: ROUGE-2 and LaSE scores for Hindi, Arabic, and Russian as target pivots as the sources languages vary. Just like Figure 4, the m2m model significantly outperforms the m2o models and s. + t. baseline on most languages.
Figure 9: ROUGE-2 and LaSE scores for Hindi, Arabic, and Russian as source pivots as the target languages vary. Just like Figure 5, the m2m model significantly outperforms the o2m models and s. + t. baseline on most languages.
Figure 10: Zero-shot ROUGE-2 scores for the different target languages as the source languages vary. The zero-shot models are trained with only the in-language samples of the pivot. Though their results are clearly behind the fully supervised models, the zero-shot models are able to generate non-trivial summaries for many language pairs.
Figure 11: Zero-shot LaSE scores for the different source languages as the target languages vary. The zero-shot models are trained with only the in-language samples of the pivot. Though their results are clearly behind the fully supervised models, the zero-shot models are able to generate non-trivial summaries for many language pairs.
Table 4: An article-summary statistics of the CrossSum dataset containing a total of 1,678,466 cross-lingual samples. The rows indicate the articles’ language and columns that of their summaries’. For example, the cell on the second column of the fourth row indicates the number of samples where the article is in Bengali and the summary in Arabic.