# XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages

**Tahmid Hasan**[1*]**, Abhik Bhattacharjee**[1*]**, Md Saiful Islam**[2]**, Kazi Samin**[1]**,**
**Yuan-Fang Li**[3]**, Yong-Bin Kang**[4]**, M. Sohel Rahman**[1]**, Rifat Shahriyar**[1]

Bangladesh University of Engineering and Technology (BUET)[1], University of Rochester[2],
Monash University[3], Swinburne University of Technology[4]

tahmidhasan@cse.buet.ac.bd, {abhik,samin}@ra.cse.buet.ac.bd,
mislam6@ur.rochester.edu, yuanfang.li@monash.edu,
ykang@swin.edu.au, {msrahman,rifat}@cse.buet.ac.bd

## Abstract

Contemporary works on abstractive text summarization have focused primarily on high-resource languages like English, mostly due to the limited availability of datasets for low/mid-resource ones. In this work, we present XL-Sum, a comprehensive and diverse dataset comprising 1 million professionally annotated article-summary pairs from BBC, extracted using a set of carefully designed heuristics. The dataset covers 44 languages ranging from low to high-resource, for many of which no public dataset is currently available. XL-Sum is highly abstractive, concise, and of high quality, as indicated by human and intrinsic evaluation. We fine-tune mT5, a state-of-the-art pretrained multilingual model, with XL-Sum and experiment on multilingual and low-resource summarization tasks. XL-Sum induces competitive results compared to the ones obtained using similar monolingual datasets: we show higher than 11 ROUGE-2 scores on 10 languages we benchmark on, with some of them exceeding 15, as obtained by multilingual training. Additionally, training on low-resource languages individually also provides competitive performance. To the best of our knowledge, XL-Sum is the largest abstractive summarization dataset in terms of the number of samples collected from a single source and the number of languages covered. We are releasing our dataset and models to encourage future research on multilingual abstractive summarization. The resources can be found at https://github.com/csebuetnlp/xl-sum.

## 1 Introduction

Automatic text summarization (Nenkova and McKeown, 2011) is a fundamental problem in natural language processing. Given an input text (typically a long document or article), the goal is to

---

*These authors contributed equally to this work.

| | |
|---|---|
| **Input Article:** Yahoo's patents suggest users could weigh the type of ads against the sizes of discount before purchase. It says in two US patent applications that ads for digital book readers have been "less than optimal" to date. [...] "Greater levels of advertising, which may be more valuable to an advertiser and potentially more distracting to an e-book reader, may warrant higher discounts," it states. [...] It adds that the more willing the customer is to see the ads, the greater the potential discount. [...] At present, several Amazon and Kobo e-book readers offer full-screen adverts when the device is switched off and show smaller ads on their menu screens. [...] Yahoo does not currently provide ads to these devices, and a move into the area could boost its shrinking revenues. | |
| **Summary:** Yahoo has signalled it is investigating e-book adverts as a way to stimulate its earnings. | |

Table 1: A sample article-summary pair from the XL-Sum dataset. To highlight the abstractiveness of the summary, we underline the novel words and phrases that do not appear in the article text. Also, portions of the article relevant to the summary have been color-coded. As we can see, these portions are concisely paraphrased in the summary, unlike extractive methods.

generate a smaller, concise piece of text that conveys the key information of the input text. There are two main approaches to automatic text summarization: *extractive* and *abstractive*. Extractive methods crop out one or more segments from the input text and concatenate them to produce a summary. These methods were dominant in the early era of summarization, but they suffer from some limitations, including weak coherence between sentences, inability to simplify complex and long sentences, and unintended repetition (See et al., 2017; Widyassari et al., 2020).

Abstractive summarization, on the other hand, generates summaries that may contain words and phrases not present in the input text (e.g., via paraphrasing), and may arguably relate more to human-generated summaries (Hsu et al., 2018). Although abstractive summaries can be more coherent and concise than extractive summaries (Cohn and Lapata, 2008), generating them is more challenging due to the nature of this task. Limited availability of good datasets conducive to abstractive methods has made it even more difficult. For these reasons, extractive models had been performing better than abstractive ones historically. However, the success of sequence-to-sequence (seq2seq) models (Cho et al., 2014; Sutskever et al., 2014) over the last decade and the recent advances in Transformer-based models (Vaswani et al., 2017; Devlin et al., 2019) have rejuvenated abstractive text summarization (Rush et al., 2015; See et al., 2017; Zhang et al., 2020), which had previously received much less attention in comparison to extractive approaches (Nenkova and McKeown, 2012). Still, the scarcity of good datasets, especially for low-resource languages, remains a roadblock.

Typical seq2seq models are heavily data-driven, i.e., a large number of article-summary pairs are required to train them effectively. As a result, abstractive summarization has centered around the English language, as most large abstractive summarization datasets (Hermann et al., 2015; Grusky et al., 2018; Narayan et al., 2018) are available in English only. Though there have been some recent efforts for curating multilingual abstractive summarization datasets (Giannakopoulos et al., 2015; Cao et al., 2020; Scialom et al., 2020), they are limited in terms of the number of languages covered, the number of training samples, or both.

In this work, we introduce **XL-Sum**, a large-scale abstractive summarization dataset of news articles crawled from the British Broadcasting Corporation (BBC)[1] website. Using a custom crawler, we collect 1 million professionally annotated article-summary pairs covering 44 languages. Originating from a single source, these samples exhibit similar summarization strategies across all languages, making them ideal for the multilingual summarization task. XL-Sum introduces the first publicly available summarization dataset and benchmarks for many languages

(e.g., Bengali, Swahili). Thus, this dataset potentially enables and facilitates research on low-resource languages, bringing technological advances to communities of these languages that have been traditionally under-served.

We achieve higher than 11 ROUGE-2 score on the 10 languages we benchmark on multilingual summarization, even exceeding 15 ROUGE-2 score (16.58 being the state-of-the-art for English, obtained by Zhang et al. (2020) on XSum (Narayan et al., 2018), a similar dataset) on many of them. In addition, we also conduct experiments on low-resource summarization task and show competitive results, indicating that the dataset can be used even for the low-resource languages individually.

In summary, we make the following main contributions in this paper:

- We release XL-Sum, a dataset containing 1 million article-summary pairs in 44 languages, being the first publicly available abstractive summarization dataset for many of them.

- We create a data curation tool that can automatically crawl and extract article-summary pairs from BBC, using which the dataset can be made even larger over time.

- We are the first to perform multilingual summarization on a diverse set of languages, achieving strong baselines on all languages tested.

We are releasing the dataset, curation tool, and summarization model checkpoints. We believe that our efforts in this work will encourage the community to push the boundaries of abstractive text summarization beyond the English language, especially for low and mid-resource languages.

## 2 The XL-Sum Dataset

In this section, we present details of the XL-Sum dataset together with the curation process. Table 2 shows the article-summary statistics for all languages in the XL-Sum dataset.

### 2.1 Content Source

BBC publishes news in 43 languages[2] ranging from low-resource languages such as Bengali and

---

| Language | #Samples | Language | #Samples | Language | #Samples |
|----------|---------|----------|---------|----------|---------|
| Amharic | 5,461 | Korean | 4,281 | Somali | 5,636 |
| Arabic | 40,327 | Kyrgyz | 2,315 | Spanish | 44,413 |
| Azerbaijani | 7,332 | Marathi | 11,164 | Swahili | 10,005 |
| Bengali | 8,226 | Nepali | 5,286 | Tamil | 17,846 |
| Burmese | 5,002 | Oromo | 5,738 | Telugu | 11,308 |
| Chinese | 39,810 | Pashto | 15,274 | Thai | 6,928 |
| English | 301,444 | Persian | 25,783 | Tigrinya | 4,827 |
| French | 9,100 | Pidgin[a] | 9,715 | Turkish | 29,510 |
| Gujarati | 9,665 | Portuguese | 23,521 | Ukrainian | 57,952 |
| Hausa | 6,313 | Punjabi | 8,678 | Urdu | 40,714 |
| Hindi | 51,715 | Russian | 52,712 | Uzbek | 4,944 |
| Igbo | 4,559 | Scottish Gaelic | 1,101 | Vietnamese | 23,468 |
| Indonesian | 44,170 | Serbian (Cyrillic) | 7,317 | Welsh | 11,596 |
| Japanese | 7,585 | Serbian (Latin) | 7,263 | Yoruba | 6,316 |
| Kirundi | 5,558 | Sinhala | 3,414 | **Total** | **1,005,292** |

Table 2: Languages covered by the XL-Sum dataset, and the number of samples for each language. Here, a sample denotes an article-summary pair. If we consider languages with less than 15,000 training samples to be low-resource, then more than two-thirds of the constituent languages in XL-Sum fall into this category.

---

[a] West African Pidgin English

Swahili to high-resource ones including English and Russian. Among the 43 languages, Serbian is a special case that is published in both Cyrillic, the official script, and Latin, the colloquial script. We treat them as different languages in this work, totaling to a coverage of 44 languages.

## 2.2 Content Search

As BBC does not provide any archive or RSS feed on their website, we designed a crawler to recursively crawl pages starting from the homepage by visiting different article links present in each page visited. We were able to take advantage of the fact that all BBC sites have somewhat similar structures, and were able to scrape articles from all sites. We discarded pages with no textual contents (mostly pages consisting of multimedia contents) before further processing.

## 2.3 Article-Summary Extraction

The process of automatically collecting summaries of articles differs across different datasets. For example, the CNN/DM dataset (Hermann et al., 2015) merged bullet point highlights provided with the articles as reference summaries, whereas the XSum dataset (Narayan et al., 2018) used the first line of the article as the summary and the rest of the article as the input.

Our method of collecting summaries was made

easier by the consistent editorial style of the BBC articles we crawled. BBC typically provides a summary of a whole article in the form of a bold paragraph containing one or two sentences at the beginning of each article. These summaries are written professionally by the authors of the articles in order to convey its main story within one small paragraph. This is in contrast to the headline which serves to draw the attention of viewers into reading the article. (We show an example article-summary pair from BBC English in Table 1.) We designed a number of heuristics to make the extraction effective by carefully examining the HTML structures of the crawled pages:

1. The desired summary must be present within the beginning two paragraphs of an article.

2. The summary paragraph must have some portion of texts in bold format.

3. The summary paragraph may contain some hyperlinks that may not be bold. The proportion of bold texts and hyperlinked texts to the total length of the paragraph in consideration must be at least 95%.

4. All texts except the summary and the headline must be included in the input text (including image captions).

5. The input text must be at least twice as large as the summary.

Any sample that did not conform to these heuristics were discarded. Our strategy of automatic annotation of summaries resembles XSum to some extent, but we found the first line to contain meta-information in many articles (e.g., author information, date of last modification). As such, we used the bold paragraphs as the summaries instead.

## 3  Human Evaluation of XL-Sum

Despite being written by professionals, evaluating the quality of the XL-Sum dataset is crucial to ensure that it can be valuable and used by a wider community for abstractive summarization. For this, we thoroughly conducted human evaluations on a subset of the dataset.

We hired professional annotators[3] to assess the quality of the top-10 languages according to the number of speakers worldwide.[4] It is worth noting that not all of these 10 languages are high-resource (e.g., Bengali is a low-resource language despite being one of the most widely spoken).

Each evaluator was asked to assess the quality of a random subset of the dataset (around 250 article-summary pairs) by answering the following questions with 'Yes'/'No':

**Property A:** Does the summary convey what the article is about?

**Property B:** If the answer to property A is 'Yes', does the summary contain any information that is inconsistent with the article?

**Property C:** If the answer to property A is 'Yes', does the summary contain any information that cannot be inferred from the article?

The motivation for designing these properties stemmed from recent progress on the quality estimation of neural language generation (NLG) models. Belinkov and Bisk (2018) showed that NLG models are vulnerable to noisy and low-quality training samples, hence, it is essential to validate the quality of the summary through Property A. Ensuring factual consistency and faithfulness (Wang et al., 2020; Maynez et al., 2020) of the generated summaries is also a key consideration

for neural abstractive summarization since neural models have been shown to generate hallucinated texts (See et al., 2017). Property B checks for consistency between the article and the summary; while Property C assesses the hallucination implicitly by limiting the knowledge domain to the input article and identifying additional information present in the summary.

| Language/Dataset | A | B | C |
|---|---|---|---|
| CNN/DM | 98.33 | 1.22 | 24.57 |
| XSum | 92.00 | 0.00 | 71.74 |
| English | 99.66 | 0.00 | 37.37 |
| Chinese | 93.49 | 0.00 | 29.56 |
| Hindi | 90.91 | 0.00 | 31.42 |
| Spanish | 84.71 | 0.00 | 42.93 |
| French | 99.20 | 0.00 | 26.72 |
| Arabic | 98.34 | 0.00 | 25.31 |
| Bengali | 91.14 | 0.00 | 26.85 |
| Russian | 95.65 | 0.00 | 38.64 |
| Portuguese | 88.31 | 0.47 | 38.50 |
| Indonesian | 97.59 | 0.41 | 27.57 |

Table 3: Quality of XL-Sum (along with CNN/DM and XSum) assessed by human evaluators. In most of the cases, the evaluators agree that the summaries convey the main idea (Property A), and do not conflict with the input text (Property B). However, the summaries may contain some additional information (Property C), since the editors writing the summaries may use their common sense and additional domain knowledge not present in the article.

It is desirable that property A would have a high percentage of 'Yes', while both property B and C would have a low percentage. However, the third property can be a bit ambiguous, since some pieces of information may not be directly present in the input article, but a professional with background knowledge and understanding of the article topic may infer them inherently. Since text summarization is posed as a closed-domain task, we asked the annotators to label accordingly, i.e., not making use of any additional information outside the article. We provided them with some sample annotations in English to aid them during annotation. The percentages of 'Yes' for the three properties are shown in Table 3.[5] We also showed human evaluation of CNN/DM and XSum for contrast. Each article-summary pair was labeled by

---

[3]Each evaluator had at least an undergraduate degree and had native or bilingual proficiency in the languages they were assigned to. Bilingual proficiency was determined by professional language tests for the corresponding language.

[4]https://w.wiki/Pss

[5]We measured inter-annotator agreement using Cohen's kappa coefficient. Most scores were within the 0.7-0.9 range, showing high agreement between the evaluators

two separate evaluators, and we considered each pair to exhibit property A if both agreed upon it and property B and C if at least one of them did.

We can see from Table 3 (Property A) that most languages showed a high percentage of good summaries in the upper nineties, while some had a slightly lower percentage (i.e., Spanish and Portuguese). We were informed by the annotators that the negative summaries were mostly extracted from opinion pieces and blog posts, the bold paragraphs of which did not convey the main story of the articles.

Almost none of the summaries contained any conflicting information (Property B), while about one-third on average had information that was not directly inferrable (Property C). Interestingly, more than 75% of the latter had missing information such as first names, designations, or elaborations of acronyms. For example, a summary had the phrase 'President-elect Joe Biden' while its corresponding article had no presence of the first name 'Joe'. Another article had NHS elaborated in the summary, which was absent in the article. In general. the types of extra information carried in the summaries were more or less the same across all languages. CNN/DM and XSum exhibit extra information in the summaries as well, implying this phenomenon is common in abstractive text summarization datasets.

The presence of extra information in the summaries is understandable, since professional experts writing these summaries not only use the information present in the article text, but also incorporate their knowledge and understanding of the outside world. But for a closed-domain summarization model or a layman to the topic, inferring this information is not so straightforward, which makes the automatic abstractive summarization task more challenging. This phenomenon may explain why language models fine-tuned on pre-trained checkpoints (Raffel et al., 2020; Qi et al., 2020; Zhang et al., 2020) are achieving state-of-the-art results in abstractive summarization, as they are able to make use of outside information gained from the high volume of texts they were pretrained with. Additionally, it would be interesting to investigate whether the recent trend of incorporating real-world knowledge and commonsense reasoning (Tandon et al., 2018; Deng et al., 2020) into language models could improve text summarization performance.

## 4 Intrinsic Evaluation of XL-Sum

Although the human evaluations provided good insights into the quality of the summaries, there are several other aspects of the summaries that are often infeasible or impractical to judge by human evaluators. With the above backdrop, several works (Narayan et al., 2018; Grusky et al., 2018; Bommasani and Cardie, 2020) have proposed many automatic metrics to quantify important features of abstractive summaries (e.g., novel words, abstractivity, compression, and redundancy).

**Novel n-gram ratio:** Narayan et al. (2018) proposed the percentage of n-grams in the summary that do not occur in the input article as a means of measuring abstractiveness.

**Abstractivity:** Grusky et al. (2018) introduced *fragments*, which greedily match text spans between the article and the summary, and Bommasani and Cardie (2020) generalized it to introduce *abstractivity* to measure absractiveness.

**Compression:** Bommasani and Cardie (2020) proposed *compression* as a metric to quantify conciseness. Compression is measured by

$$\mathbf{CMP}(A, S) = 1 - \frac{|S|}{|A|} \qquad (1)$$

where $|A|$ and $|S|$ denote the length of the article and the summary, respectively. We measured length in terms of number of tokens.

**Redundancy:** Although Bommasani and Cardie (2020) proposed a metric to measure *redundancy*, it is only applicable to multi-sentence summaries, which is not the case for most examples in XL-Sum. Thus, we propose a new redundancy metric by calculating the number of repetitive n-grams in the summary text.

Let $\{g_1, g_2, \cdots, g_m\}$ be the unique n-grams occurring in a summary $S$, and let $\{f_1, f_2, \cdots, f_m\}$ be their respective frequencies. Then the total number of repeated n-grams are $\sum_{i=1}^{m}(f_i - 1)$. We define redundancy as the ratio of redundant n-grams and the total number of n-grams in $S$:

$$\mathbf{RED}(S) = \frac{\sum_{i=1}^{m}(f_i - 1)}{\sum_{i=1}^{m} f_i}$$
$$= 1 - \frac{m}{|S| - n + 1} \qquad (2)$$

It is preferred of a good summary to have a high novel n-gram ratio, abstractivity, and compression; while having a low redundancy score.

| Language /Dataset | Percentage of novel n-grams ↑ | | | | ABS ↑ | CMP ↑ | RED (n=1) ↓ | RED (n=2) ↓ |
|---|---|---|---|---|---|---|---|---|
| | n = 1 | n = 2 | n = 3 | n = 4 | | | | |
| CNN/DM | 13.20 | 52.77 | 72.22 | 81.40 | 38.75 | 90.90 | 13.73 | 1.10 |
| XSum | 35.76 | 83.45 | 95.50 | 98.49 | 75.70 | 90.40 | 5.83 | 0.16 |
| English | 32.22 | 80.99 | 94.57 | 98.06 | 71.74 | 92.97 | 6.56 | 0.20 |
| Chinese | 36.13 | 79.23 | 91.14 | 94.58 | 70.23 | 92.95 | 7.37 | 0.50 |
| Hindi | 29.55 | 74.77 | 90.87 | 96.29 | 64.63 | 93.00 | 9.91 | 0.16 |
| Spanish | 32.63 | 76.29 | 91.96 | 96.57 | 66.60 | 92.49 | 11.45 | 0.57 |
| French | 35.41 | 74.72 | 88.39 | 93.24 | 65.29 | 88.34 | 8.34 | 0.44 |
| Arabic | 49.88 | 84.56 | 94.79 | 98.10 | 76.72 | 90.62 | 3.93 | 0.18 |
| Bengali | 38.81 | 81.10 | 92.10 | 95.89 | 72.76 | 94.74 | 2.93 | 0.25 |
| Russian | 49.27 | 85.89 | 95.57 | 98.34 | 78.39 | 91.25 | 4.34 | 0.16 |
| Portuguese | 30.28 | 77.11 | 92.23 | 96.71 | 66.80 | 94.47 | 10.22 | 0.34 |
| Indonesian | 33.54 | 76.87 | 91.73 | 96.53 | 66.68 | 91.62 | 3.94 | 0.23 |

Table 4: Intrinsic evaluation of our XL-Sum dataset compared to CNN/Daily Mail and XSum. All values are reported in percentage for easier comparison. We use ↑ to indicate "higher is better" and ↓ for the reverse. Both of XL-Sum and XSum are highly abstractive, concise, and shows comparable quality, although the XSum dataset contains only English samples. For both XL-Sum and XSum, percentages of novel n-grams (n = 1, 2, 3, 4) are significantly higher than CNN/DM. High abstractiveness (ABS) scores of XL-Sum and XSum also bolster this finding. Additionally, low redundancy (RED) and high compression (CMP) values indicate that XL-Sum and XSum are more concise than CNN/DM.

We show these metrics in Table 4 (for redundancy, we report values for $n = 1, 2$). We also show these metrics for the CNN/DM and XSum datasets.

The results indicate that the XL-Sum dataset is highly abstractive—about one-third of the tokens, and more than 75% of the bigrams in the summaries are novel, and the abstractiveness score is also high (more than 65% for most of the languages). Additionally, XL-Sum is very concise (the summary is less than one-tenth of the input article for most of the languages), and contains minimal redundancy (less than 10% for the majority). The quality of XSum is comparable, however, it is limited to only one language (i.e., English). On the other hand, the CNN/Daily Mail dataset is significantly behind XL-Sum and XSum as indicated by most of the metrics mentioned above.

## 5 Experiments and Benchmarks

In previous sections, we have discussed the quality of XL-Sum. In addition, it is imperative to see how state-of-the-art models perform when trained on this dataset. Moreover, for many languages (e.g., Bengali, Swahili), currently, there is no publicly available dataset and benchmarks for abstractive text summarization to the best of our knowledge. In this section, we train summarization models with the XL-Sum dataset and provide several baselines and benchmark results.

Fine-tuning Transformer-based (Vaswani et al., 2017) seq2seq models initialized with pretrained weights from self-supervised training (Raffel et al., 2020; Liu and Lapata, 2019; Rothe et al., 2020; Qi et al., 2020; Zhang et al., 2020) has been shown to achieve state-of-the-art performance on many abstractive text summarization datasets. There are many multilingual pretrained checkpoints available through the Hugging Face Transformers Library (Wolf et al., 2020). Among them, we chose to use the mT5 model (Xue et al., 2021), a multilingual language model pretrained on a large dataset of 101 languages.

We performed summarization experiments in two settings: (i) multilingual, and (ii) low-resource. For performance reporting, for each language, we randomly sampled 500 pairs for its development set and 500 pairs for its test set, while using the rest of the pairs for training. We tokenized our training samples using the 250k wordpiece (Wu et al., 2016) vocabulary provided with the mT5 checkpoint. Due to computational constraints, we used the base model (600M parameters) and had to truncate the inputs to 512 tokens and the outputs to 64 tokens. We used the ROUGE-1, ROUGE-2 and ROUGE-L (Lin, 2004) scores for automatic evaluation. For inference, we used beam search with beam size 4 and length penalty of $\alpha = 0.6$ (Wu et al., 2016).

## 5.1 Multilingual Summarization

Multilingual training is performed by training a single model with training samples from multiple languages. It has been previously used in several NLP tasks including neural machine translation (Arivazhagan et al., 2019) and language model pretraining (Conneau and Lample, 2019). However, multilingual training in the context of abstractive summarization has not been a major focus of the community. As such, the aim of this experiment is to demonstrate that a single model can perform well on summarizing texts in different languages, and that sister languages with morphological similarity can take advantage of positive transfer from each other which is not possible in monolingual settings.

For this experiment, we followed a similar training strategy as Conneau and Lample (2019): we sampled each batch from a single language containing 256 samples and used a smoothing factor ($\alpha$) of 0.5 so that batches of low-resource languages would be sampled at a higher rate, increasing their frequency during training.

We fine-tuned the mT5 model for 35k steps on a distributed cluster of 8 Nvidia Tesla P100 GPUs for 4 days. We used the Adafactor optimizer (Shazeer and Stern, 2018) with a linear warmup of 5,000 steps and 'inverse square root' learning rate schedule. We show the ROUGE scores achieved by the model on the top-10 languages in Table 5.

| Language | R-1 | R-2 | R-L |
|---|---|---|---|
| English | 36.99 | 15.18 | 29.64 |
| Chinese | 36.89 | 15.23 | 30.52 |
| Hindi | 34.51 | 13.55 | 28.23 |
| Spanish | 30.93 | 12.14 | 23.76 |
| French | 34.47 | 15.94 | 27.53 |
| Arabic | 33.23 | 13.74 | 27.84 |
| Bengali | 28.32 | 11.43 | 24.23 |
| Russian | 31.10 | 13.47 | 25.54 |
| Portuguese | 31.06 | 11.62 | 23.39 |
| Indonesian | 36.17 | 16.70 | 30.50 |

Table 5: ROUGE scores for multilingual summarization achieved by the mT5 model when fine-tuned on the XL-Sum training set.

As we can see from the table, the multilingual model achieved higher than 11 ROUGE-2 scores on all languages. Some of these languages (e.g., Bengali) are low-resource, but the model still obtained competitive results comparable to high and mid-resource languages. Also, we are the first to report the abstractive summarization benchmark for a number of languages, including Bengali.

The mT5-base model achieves a R2-score of 15.18 on the English language. In comparison, the state-of-the-art PEGASUS$_{BASE}$ model (Zhang et al., 2020) obtained an R-2 score of 16.58 trained on the XSum English dataset, which is similar to XL-Sum in nature. This result suggests that the performance is comparable to the state-of-the-art English summarization. The R-2 scores for other languages are also similar to English, indicating that our dataset can help to effectively generate automatic summaries for all languages tested, including those low-resource ones.

## 5.2 Low-Resource Summarization

We have shown the effectiveness of the multilingual training strategy in summarizing articles for a wide set of languages with a single model. However, training the model is compute-heavy and it may not be realistic in many scenarios. For the dataset to be termed as versatile, models trained on individual languages should be able to perform competitively with the multilingual model. To confirm this is indeed the case, we performed training on five low-resource languages from Table 2 (Amharic, Azerbaijani, Bengali, Japanese, Swahili) in a compute-efficient setup. We fine-tuned mT5 on each language separately for 6-10 epochs (since the total training samples were limited, we had to be careful to prevent overfitting) on a single-GPU (Nvidia RTX 2080Ti) machine. For these experiments, we used a batch size of 32 and trained with a slanted learning rate schedule (Howard and Ruder, 2018). We show the ROUGE scores of each model in Table 6. We use the results from the multilingual models as baseline.

| Language | Low-resource R-1/R-2/R-L | Multilingual R-1/R-2/R-L |
|---|---|---|
| Amharic | 15.33/5.12/13.85 | **17.49/6.34/15.76** |
| Azerbaijani | 16.79/6.94/15.36 | **19.29/8.20/17.62** |
| Bengali | 25.33/9.50/22.02 | **28.32/11.43/24.02** |
| Japanese | 44.55/21.35/34.43 | **47.17/23.34/36.20** |
| Swahili | 34.29/15.97/28.21 | **38.18/18.16/30.98** |

Table 6: Performance of mT5 model fine-tuned on a low-resource training setup vs multi-lingual setup as mentioned in the previous section.

As evident by the results from Table 6, the multilingual model outperformed all the models

trained on a single language. This is expected since similar languages can have positive transfer between them (Conneau et al., 2020) when trained together. However, the low-resource models do not trail by a large margin, in all cases the difference is not more than 2 for R-2 scores. This is a good indication that models fine-tuned on such a low amount of samples can still generalize to produce results competitive to multilingual models.

The case for Amharic, Azerbaijani and Japanese call for a discussion on their performance. The first two had comparatively low scores, while the last one (Japanese) had considerably higher scores compared to the other languages. Amharic and Azerbaijani had approximately 4k and 6k training samples respectively, which we conjecture is the primary reason behind their underperformance. Moreover, we did not find any reliable stemmer to preprocess the generated summaries before computing ROUGE, which may also hurt the scores. On the other hand, Japanese texts are not word-segmented and the words need to be separated before calculating ROUGE. We used Fugashi (McCann, 2020), and possibly due to its aggressive segmentation, the scores turned out to be higher than other languages. Similar high results have also been reported while measuring BLEU (Papineni et al., 2002) scores for machine translation evaluation in Japanese (Kudo, 2018).

Results in Table 6 show that although these languages are low-resource, the scores of the two setups are close, indicating our dataset can also be useful when used with a constrained computation capability. This is likely to contribute towards advances in low-resource text summarization, enabling fairness and access to the underserved communities.

## 6 Related Works

Rush et al. (2015); Nallapati et al. (2016) pioneered neural abstractive summarization, using recurrent attentional seq2seq models (Bahdanau et al., 2015). See et al. (2017) introduced Pointer-Generator networks for abstractive summarization, which can learn to copy words from the input text, in addition to generating new texts with the decoder. Gehring et al. (2017) proposed convolutional seq2seq models and applied it to perform abstractive summarization. Narayan et al. (2018) extended the work by integrating topic embeddings with the model.

Pretrained language models have recently been successfully applied to abstractive summarization. Liu and Lapata (2019) initialized the encoder and Rothe et al. (2020) initialized both the encoder and the decoder of a seq2seq model with the pretrained BERT (Devlin et al., 2019) weights and fine-tuned the models for abstractive summarization. Raffel et al. (2020); Qi et al. (2020) used fully pre-trained seq2seq models, while Zhang et al. (2020) introduced a summarization-specific pretraining objective to achieve state-of-the-art results on multiple datasets.

Most works on abstractive summarization have so far focused on English, in large part due to a lack of benchmark datasets for other languages. Giannakopoulos et al. (2015) introduced MultiLing 2015, a summarization dataset spanning 40 languages. However, MultiLing 2015 is limited in size, with the training set having only 10k samples in total. Cao et al. (2020); Scialom et al. (2020) introduced two new datasets for multilingual summarization, but both were limited to less than 10 languages. Moreover, samples for different languages were collected from different sources, making them exposed to different types of summarization strategies, which raises questions on the uniformity of the summaries.

## 7 Conclusion and Future Works

In this paper, we present XL-Sum, a large-scale, high-quality multilingual text summarization dataset, containing 1 million samples across 44 languages collected from a single source, BBC. For many of the languages, XL-Sum provides the first publicly available abstractive summarization dataset and benchmarks. We also make the dataset curation tool available for the researchers, which will help to grow the dataset over time. Thorough human and intrinsic evaluations indicate that the summaries in our dataset are highly abstractive and concise while conveying the main idea with little to no conflict with the input article. Additionally, we demonstrate that multilingual training can help towards better summarization, most likely due to the positive transfer between sister languages with morphological similarity. Moreover, XL-Sum is also useful in a low-resource and compute-efficient setting.

In future, we will investigate the use of our dataset for other summarization tasks (e.g., cross-lingual summarization Zhu et al., 2019).

We hope the XL-Sum dataset will be helpful for the research community, especially for the researchers working to ensure fair access to technological advances to under-served communities with low-resource languages.

## Acknowledgements

## References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, California, USA.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, BC, Canada.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Yue Cao, Xiaojun Wan, Jinge Yao, and Dian Yu. 2020. Multisumm: Towards a unified model for multilingual abstractive summarization. In *Proceedings of Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 11–18. AAAI Press.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK.

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 7059–7069, Vancouver, Canada.

Changyu Deng, Xunbi Ji, Colton Rainey, Jianyu Zhang, and Wei Lu. 2020. Integrating machine learning with human knowledge. *iScience*, 23(11):101656.

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, USA. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70, pages 1243–1252. PMLR.

George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. MultiLing 2015: Multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–274, Prague, Czech Republic. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings*

of the 28th International Conference on Neural Information Processing Systems (NIPS 2015), pages 1693–1701, Montreal, Canada.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online. Association for Computational Linguistics.

Paul McCann. 2020. fugashi, a tool for tokenizing Japanese in python. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), pages 44–51, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. Foundations and Trends® in Information Retrieval, 5(2–3):103–233.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In Mining text data, pages 43–76. Springer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2401–2410, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21:1–67.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics, 8:264–280.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8051–8067, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, pages 4603–4611. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014)*, pages 3104–3112, Montreal, Canada.

Niket Tandon, Aparna S Varde, and Gerard de Melo. 2018. Commonsense knowledge in machine intelligence. *ACM SIGMOD Record*, 46(4):49–52.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, page 6000–6010, Long Beach, California, USA.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, et al. 2020. Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*.

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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119, pages 11328–11339. PMLR.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. 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 3054–3064, Hong Kong, China. Association for Computational Linguistics.