M.Sc. Engg. (CSE) Thesis

Better Conditional Text Generation with Low-Resource and Multilingual Models

Submitted by

Tahmid Hasan

0419052048

Supervised by

Dr. Rifat Shahriyar

Submitted to

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering

May 2022

Candidate's Declaration

I, do, hereby, certify that the work presented in this thesis, titled, "Better Conditional Text Generation with Low-Resource and Multilingual Models", is the outcome of the investigation and research carried out by me under the supervision of Dr. Rifat Shahriyar, Professor, Department of CSE, BUET.

I also declare that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Tahmid Hasan
0419052048

The thesis titled "**Better Conditional Text Generation with Low-Resource and Multilingual Models**", submitted by Tahmid Hasan, Student ID 0419052048, Session April 2019, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents on May 14, 2022.

Board of Examiners

1. Rifat

Dr. Rifat Shahriyar — Chairman
Professor — (Supervisor)
Department of CSE, BUET, Dhaka

2.

Dr. Mahmuda Naznin — Member
Professor and Head — (Ex-Officio)
Department of CSE, BUET, Dhaka

3. M. M. Islam

Dr. Md. Monirul Islam — Member
Professor
Department of CSE, BUET, Dhaka

4.

Dr. A. B. M. Alim Al Islam — Member
Professor
Department of CSE, BUET, Dhaka

5.

Dr. Mohammad Nurul Huda — Member
Professor — (External)
Department of CSE
United International University, Dhaka

Acknowledgement

I am thankful to the wonderful researchers I had the good fortune to work with. In particular, I would like to express my gratitude to Abhik Bhattacharjee, a mentee turned mentor, who has shown his continuous support through my journey into research.

Finally, I would like to thank my parents for their never-ending love. Without them, my works would not have come into fruition.

Dhaka
May 14, 2022

Tahmid Hasan
0419052048

Contents

Candidate's Declaration **i**

Board of Examiners **ii**

Acknowledgement **iii**

List of Figures **vi**

List of Tables **vii**

ix

Abstract **x**

1 Introduction **1**

1.1 Motivations and Problem Statements 1

1.2 Objectives and Outcomes 2

1.3 Methodologies and Contributions 3

1.4 Thesis Outline 4

2 Backgrounds and Related Works **5**

2.1 Language Models 5

2.2 Neural Networks 6

2.2.1 Training Neural Networks 6

2.3 Distributed Representations of Words 8

2.4 Neural Language Models 10

2.4.1 Window-based Neural Language Models 10

2.4.2 Recurrent Neural Network-based Language Models 10

2.5 Conditional Text Generation and Sequence-to-Sequence Models 11

2.6 Transformer Language Models 12

2.7 Subword Vocabulary 13

2.8 Contextualized Word Representations 14

2.9 Pretrained Language Models 14

Multilingual Language Models 15
Low-Resource Machine Translation in the Context of Bengali 15
Multilingual Abstractive Text Summarization 16

3 Low-Resource Machine Translation 18
Sentence Segmentation 19
Aligner Selection and Ensembling 20
Aligner Descriptions 20
Aligner Ensembling and Filtering 21
Training Data and Batch Filtering 24
Sentence-aligned Corpora 25
Document-aligned Corpora 27
Batch Filtering 28
Evaluation Data 29
Experiments and Results 30
Pre-processing 30
Comparison with Previous Results 31
Comparison with Automatic Translators 31
Comparison with Human Performance 32
Ablation Study of Filtered Ensembles 32
Sample Complexity Tests 34

4 Multilingual Text Summarization 36
The XL-Sum Dataset 38
Content Source 38
Content Search 39
Article-Summary Extraction 39
Intrinsic Evaluation of XL-Sum 40
Multilingual ROUGE Scoring 42
Experiments and Benchmarks 42
Multilingual Summarization 43
Comparison with Monolingual Summarization 45

5 Limitations and Ethical Considerations 47

6 Conclusion and Future Works 49

References 51

List of Figures

Erroneous sentence segmentation by Polyglot 19

One-to-one sentence alignment 20

One-to-many sentence alignment 20

Many-to-one sentence alignment........ 20

Precision vs. Margin........ 23

Recall vs. Margin 24

F_1 Score vs. Margin........ 25

SacreBLEU vs Steps on SIPCdev set 33

SacreBLEU vs Document Count on SUPara test set 34

SacreBLEU vs Document Count on SIPC test set........ 34

4.1 A portion of an an example HTML page from BBC English News. The bold text (marked inside the blue box) is used as the summary whilst the the rest of texts (marked inside the green boxes) are used as the corresponding input article. 40

List of Tables

Performance metrics of aligners in terms of precision, recall and F_1 scores. The most representative metric, F_1 score indicates that Hunalign outperforms the other aligners. 22

Performance metrics of ensembles in terms of precision, recall and F_1 scores. Scores indicate that the BH ensemble outperforms the others while being almost identical in performance compared to the standalone aligner Hunalign due to the drop in prescision. 22

Performance metrics of filtered ensembles in terms of precision, recall and F_1 scores. Scores indicate that the BH ensemble with a filter of margin 0.96 applied outperforms the other standalone aligners, ensembles, and filtereed ensembles. 22

Summary of the training corpus we curated using the proposed aligner ensembling and batch filtering algorithm. We mention the number of parallel sentences, total number of tokens in Bengali and English, and per sentence average token count in Bengali and English. 26

Filtered pairs (%) for different neighborhoods. Here, document-level filtering results in the fewest number of sentence pairs filtered out. However, the 1k size batch filter remains very close within a 1-2% margin. 29

Comparison (BLEU) with previous works on SUPara-benchmark test set (Hasanet al. [42] and Alam et al. [5] did not provide En→Bn scores). Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56]. 31

Comparison (SacreBLEU) with automatic translators. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56]. 32

SacreBLEU scores for ablation study. SacreBLEU scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56]. 32

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 10,000 training samples to be low-resource, then more than half of the constituent languages in XL-Sum fall into this category 37

A sample article-summary pair from the XL-Sum dataset. To highlight the abstractiveness of the summary, we underline the novel words and phrases thatdo 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... 38

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................41

ROUGE scores for multilingual summarization achieved by the mT5 model when fine-tuned on the XL-Sum training set. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56]............ 44

Performance of mT5 model fine-tuned on a low-resource training setup vs multilingual setup as mentioned in the previous section. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56]... 45

List of Algorithms

1 A pseudocode of the aligner ensembling and batch filtering algorithm. 29

2 A pseudocode of the sampling algorithm for multilingual training. 43

Abstract

Recent advances in deep learning have aided in the development of neural language models that have achieved state-of-the-art results in many natural language processing (NLP) tasks. Conditional text generation, a major subfield of NLP, has particularly benefited from neural sequence-to-sequence (seq2seq) models, which can generate a text sequence when conditioned on a given input text sequence. These seq2seq models, however, come with a major caveat: they tend to be heavily data-driven, i.e., a large number of training samples must be fed into these models to train them effectively, and the absence of which can even affect their performance substantially. This has thus limited the applicability of these models to only the languages for which there are large datasets available, i.e., the high-resource languages. As a result, low-resource languages (e.g., Bengali) often fail to reap the benefit of these models and trail significantly in performance compared to high-resource ones. Even in multilingual language models, which are trained on hundreds of languages, low-resource languages remain underrepresented, as they are often not the primary focus of these models. These above-mentioned effects have only cascaded and barred the advancement of major NLG applications (e.g., machine translation, text summarization) from the under-served low-resource communities. In this work, we explore two major conditional text generation problems, machine translation and abstractive text summarization, from a low-resource and multilingual perspective. We improve the sentence segmentation algorithm for Bengali and propose two novel alignment techniques and effective algorithms for parallel corpus creation for machine translation under low-resource scenarios. Side by side, we create a large parallel training corpus and establish reliable evaluation benchmarks for Bengali-English machine translation as a representative low-resource language pair. Furthermore, for the first-time ever, we introduce a set of novel automatic annotation techniques and curate a large-scale multilingual dataset for abstractive text summarization and benchmark on the multilingual summarization task using a new multilingual metric for evaluation of the model-generated summaries. We show the superiority of multilingual training over back-translation-based and monolingual summarization.

Chapter 1

Introduction

Machine translation [114] and Abstractive Text Summarization [71] are two of the most fundamental natural language processing (NLP) tasks under the umbrella of a common modeling paradigm: conditional text generation [36]. As the name suggests, given a piece of text sequence as input, conditional text generation models produce another text depending on the task at hand. While being extrinsically different, machine translation and abstractive text summarization share identical model backbones: sequence-to-sequence models [115]. Most commonly known as seq2seq models, these neural network-based models have emerged as the state-of-the-art model types for machine translation [10, 22, 118] and text summarization [99, 102, 108] over the last decade. However, as with any heavily parameterized neural model, seq2seq models come with a major drawback: they require a significant amount of labeled samples to train effectively [59]. While, in hindsight, it may seem desirable to have models that are capable of learning from large amounts of data, the benefits diminish once we shift to a low-data regime. For instance, it often becomes challenging to find labeled examples for languages with a smaller digital footprint (e.g., Bengali). It still remains an open question as to how to effectively train neural models with a limited amount of data while achieving the same generalization capabilities as those trained with ample data [20, 98]. In this thesis, we study the above-mentioned text generation tasks from a low-resource and multilingual perspective. We show that clever manipulation of data coupled with multilingual training strategies can overcome the limitations of large conditional text generation models in tackling the problem of learning with limited data. We first present our readers with the motivations and an overview of the problems we address.

Motivations and Problem Statements

With a population of more than 7 billion people, there are currently more than 7,000[1] languages in the world with numerous dialects and typologies that complement them. While language

[1] https://www.ethnologue.com/guides/how-many-languages

diversity is a human trait that should be celebrated, it often becomes a barrier to human-human communication, which is becoming more telling with the increase of globalization. Language technologies, therefore, are crucial necessities in this 21^{st} century to combat this communication barrier. Natural language processing, a branch of computer science that aims to enable computers to understand human language and, in the process, communicate with humans through natural human language, can also help humans speaking different languages communicate with each other via the task that is known as machine translation. Machine translation, a prevalent NLP task, takes inputs in one (source) language, most commonly in textual format, and generates an output in another (target) language. It is expected that the generated output conveys the same meaning as the original input while being produced in a completely different language.

The growth of the Internet and storage technologies has caused an exponential growth in data being produced. It is estimated that a total of above 1 million terabytes of data is being generated every day[2]. This is increasingly creating a challenge for us to consume and process the vast amount of data. Often we desire to know only the salient piece of information in a web page or document, be it a news article, a scientific paper, or a medical report. The branch of NLP that deals with condensing large pieces of text into a succinct and compressed digest is known as automatic text summarization. Abstractive text summarization, a more challenging type of summarization, aims to generate human-like summaries containing many novel words or phrases while at the same time, being faithful to the original content.

Objectives and Outcomes

We explore two major conditional text generation problems, machine translation, and abstractive text summarization, from a low-resource and multilingual perspective. The main objectives of our study are as follows:

1. To propose novel alignment techniques and effective algorithms for parallel corpus creation for machine translation under low-resource scenarios.
2. To create a large parallel training corpus and reliable evaluation benchmarks for Bengali-English machine translation as a representative low-resource language pair.
3. To introduce novel automatic annotation techniques and curate a large-scale multilingual dataset for abstractive text summarization for the first time ever.
4. To benchmark on the multilingual summarization task using a new multilingual metric to evaluate the model-generated summaries.

The outcomes of our study are as follows:

[2] https://commoncrawl.org/

1. Novel alignment techniques and fast algorithms for parallel corpus creation and a large parallel corpus created using these methodologies.
2. State-of-the-art Bengali-English neural machine translation models with a comprehensive comparison with existing works.
3. A large-scale, high-quality multilingual abstractive summarization dataset made with carefully designed heuristics.
4. State-of-the-art multilingual abstractive summarization models with benchmarks and baselines with a newly adapted multilingual evaluation metric.

Methodologies and Contributions

The methodologies and experiments in this thesis can be categorized into two major conditional text generation tasks:

1. **Machine Translation**: At first, we show that sentence segmentation plays a vital role in parallel corpus creation from noisy document pairs and investigate whether existing segmenters can split Bengali documents as effectively as English. We experiment with prominent sentence alignment algorithms and examine where they fail for Bengali-English sentence alignment. We then propose two novel techniques to increase aligned pairs without incurring incorrect alignments and present novel and fast filtering methods for removing incorrect alignments. Next, we search for different sources of noisy corpora to apply our proposed methodologies and create a large parallel training corpus. Finally, we train state-of-the-art machine translation models with datasets created with our proposed methods and compare them with existing works and automatic translators (e.g., Google Translate, Bing Translator). We evaluate a broad set of challenging test sets. We also systematically perform thorough ablation studies to show that our methods directly impact performance. Furthermore, we demonstrate that our proposed methodologies are sample-efficient: they perform better than baseline methods, especially when the number of training samples is limited.
2. **Text Summarization**: We propose a complete pipeline for auto-labeling of article-summary pairs by leveraging HTML page structures of web documents from the BBC News website. We design custom scrapers that can automatically detect and crawl potential articles and summaries from these news pages and carefully develop heuristics to extract and annotate article-summary pairs. Next, we evaluate the summaries' quality using established metrics and introduce new ones for attributes with no metrics available. We then perform multilingual training for the first time on a broad set of languages and

compare it with monolingual training on low-resource languages and back-translation-based baselines to demonstrate the superiority of multilingual training. Lastly, we will make model-generated summary evaluation metrics (e.g., ROUGE) compatible with languages beyond English, thereby making their usage universal, and then perform a thorough benchmark of our models and establish baselines for future extensions.

Thesis Outline

Chapter 1 gives a brief introduction to the problem statements and motivations behind tackling these problems, summarizes our main contributions, and provides a brief outline. Chapter 2 provides our readers with the necessary technical backgrounds required to dive deep into the textual representations, model architectures, training methodologies, and generation strategies for both tasks in a unified manner. It also covers a comprehensive literature review of related works. Chapter 3 covers machine translation, where we study effective sentence segmentation, introduce aligner ensembling and batch filtering for machine translation, and train state-of-the-art models with the dataset curated using our proposed methods. The following Chapter (Chapter 4) covers XL-Sum - a large-scale multilingual abstractive summarization dataset we curated, its quality evaluation, and fine-tuning of multilingual pretrained models with it to achieve strong results over baselines and monolingual training. We then discuss the limitations of our works and cover the ethical considerations for using our introduced datasets and models (Chapter 5). Finally, we wrap up our contributions in Chapter 6 with some future directions for extending our research.

Chapter 2

Backgrounds and Related Works

In this chapter, we provide technical backgrounds of neural language models, distributed and contextualized representations of words, sequence-to-sequence models, transformer-based models, pretrained models, and their multilingual training. We also explore existing works that build upon these models and methodologies.

Language Models

A statistical language model, or more commonly, a language model, computes the probability of a number of words occurring in a sequence. The probability of a sequence of m words $\{w_1, \cdots, w_m\}$ is denoted as $P(w_1, \cdots, w_m)$. Using conditional probability, this term is broken down as

$$P(w_1, \cdots, w_m) = \prod_{i=1}^{m} P(w_i|w_1 \cdots w_{i-1}) \tag{2.1}$$

Given a large text corpus, the task of language modeling is to learn these conditional probabilities to maximize the likelihood of the words in that corpus. For the sake of simplifying the modeling objective, n-gram language models have been introduced that assume that each word in a text corpus is dependent on the previous n words once they have been observed, i.e.,

$$P(w_1, \cdots, w_m) = \prod_{i=1}^{m} P(w_i|w_1 \cdots w_{i-1}) \approx \prod_{i=1}^{m} P(w_i|w_{i-n} \cdots w_{i-1}) \tag{2.2}$$

Using the maximum likelihood principle, we estimate the probabilities as

$$P(w_i|w_{i-n} \cdots w_{i-1}) = \frac{\text{count}(w_{i-n}, \cdots, w_i)}{\text{count}(w_{i-n}, \cdots, w_{i-1})} \tag{2.3}$$

Neural Networks

Artificial neural networks [6], inspired from biological neural networks, take a vector of any dimension n and perform a mapping $f : \mathrm{R}^n \rightarrow \mathrm{R}^m$ to output another vector of dimension m. It generates a hierarchy of representations through a series of linear projection layers and non-linear activation functions. Each dimension of an intermediate representation of a layer is called a neuron. Each neuron is connected to its previous layer's outputs through a vector called its weights. If a neuron has weights $w_1, \cdots w_n$ and the outputs from the previous layers are $x_1, \cdots, x_n$, then the output of the neuron is computed as

$$f(w_1x_1 + \cdots + w_nx_n + b) = f(w^Tx + b) \tag{2.4}$$

where f is the activation and b a bias term.

In practice, these weights and biases of all neurons in a layer are packed together into matrices for faster computation. Neural networks have strong representation capabilities and are proven to be universal function approximators [45]. This is why they have been successfully used to model probabilities by making the final layer outputs as probability distributions (via the softmax non-linearity function). We present a 1-hidden layer feed-forward neural network that models a probability distribution.

$$\begin{aligned}
\mathbf{x} &= \text{Input} \\
\mathbf{z} &= \mathbf{W_1x + b_1} \\
\mathbf{h} &= \text{ReLU}(\mathbf{x}) \\
\vartheta &= \mathbf{W_2h + b_2} \\
\hat{\mathbf{y}} &= \text{Softmax}(\vartheta)
\end{aligned}$$

Here x and $\hat{y}$ are the input and its corresponding output, W_1 and b_1 are the first layers's weights and biases, W_2 and b_2 are the second layers's weights and biases, h is the hidden layer output. Rectified linear unit [80] has been used as the hidden activation and softmax as the output activation.

Training Neural Networks

In the previous example, W_1, b_1, W_2, b_2 are model weights or parameters that are learned so that the neural network can estimate the probability distribution $\hat{y}$ given an input vector x. A popular algorithm for learning the parameters of a neural network is backpropagation [101], which is a gradient-based [65] learning algorithm.

The algorithm first initializes the weights randomly and then iteratively updates them using a forward and a backward pass. Typically a training dataset $\mathbf{D} = \{(x^{(1)}, y^{(1)}), \cdots, (x^{(n)}, y^{(n)})\}$ is provided where $\{y^{(1)}, \cdots, y^{(n)}\}$ are the true labels of the inputs. In each iteration, the inputs are passed through the neural network, and the model's probability outputs are compared to the true labels. A loss is computed, typically the cross-entropy loss or negative log-likelihood of the predicted probabilities with respect to the true labels. For example, if the output has m dimensions (i.e., m probability classes) then the cross-entropy loss is defined as

$$L(y, \hat{y}) = -\sum_{k=1}^{m} y_k \log(\hat{y}_k) \tag{2.5}$$

Computations up until this point is called the 'forward pass'.

In the 'backward pass', the errors are then used to compute the gradients

$$\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial b_1}, \frac{\partial L}{\partial W_2}, \frac{\partial L}{\partial b_2}$$

which are the partial derivatives of L with respect to W_1, b_1, W_2, b_2, respectively. The gradients can be interpreted as update signals the loss provides to the model parameters so that the loss can be minimized. Finally, the model parameters are updated using the gradients.

$$\begin{aligned}
\mathbf{U_1} &\leftarrow \mathbf{U_1} - \alpha \frac{\partial L}{\partial W_1} \\
\mathbf{b_1} &\leftarrow \mathbf{b_1} - \alpha \frac{\partial L}{\partial b_1} \\
\mathbf{U_2} &\leftarrow \mathbf{U_2} - \alpha \frac{\partial L}{\partial W_2} \\
\mathbf{b_2} &\leftarrow \mathbf{b_2} - \alpha \frac{\partial L}{\partial b_2}
\end{aligned}$$

Here α denotes the learning rate. The forward and backward passes are repeated until the model reaches convergence, i.e., weights can no longer be changed by the gradient updates.

Instead of computing the gradients directly, backpropagation leverages the chain rule of Calculus and computes them in an ordered manner, thereby reducing the number of computations. This makes the training computationally feasible. Let us now show how backpropagation works for the network above.

To start let us recall that ReLU(x) = max(x, 0). Hence, ReLU'(x) = sgn(x). Also the gradient of the cross-entropy loss with respect to the input to the softmax can be written as

$$\frac{\partial L}{\partial \vartheta} = (\hat{y} \_ y)^T \tag{2.6}$$

.

We now decompose $\frac{\partial L}{\partial W_2}$ and $\frac{\partial L}{\partial b_2}$ using the chain rule:

$$\frac{\partial L}{\partial W_2} = \frac{\partial L \, \partial \hat{y} \, \partial \vartheta}{\partial \hat{y} \, \partial \vartheta \, \partial W_2} = (\hat{y} - y)^T h^T \tag{2.7}$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L \, \partial \hat{y} \, \partial \vartheta}{\partial \hat{y} \, \partial \vartheta \, \partial b_2} = (\hat{y} - y)^T \tag{2.8}$$

Here the repetition of the computation of $\frac{\partial L}{\partial \vartheta}$ is prevented using computation graphs.

Let us now compute the intermediate gradient $\frac{\partial L}{\partial z}$:

$$\frac{\partial L}{\partial z} = \frac{\partial L \, \partial \vartheta \, \partial h}{\partial \vartheta \, \partial h \, \partial z} = (\hat{y} - y)^T W_2 \circledS \, \mathrm{sgn}(z)$$

Here $\circledS$ denotes element-wise multiplication. Now we compute $\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial b_1}$:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L \, \partial z}{\partial z \, \partial W_1} = (\hat{y} - y)^T W_2 \circledS \, \mathrm{sgn}(z) x^T \tag{2.9}$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L \, \partial z}{\partial z \, \partial b_1} = (\hat{y} - y)^T W_2 \circledS \, \mathrm{sgn}(z) \tag{2.10}$$

We can notice that each layer reuses the gradients computed in the following layer to reduce computation. Due to this computational efficiency, backpropagation has been widely adopted as the de facto algorithm for training neural networks.

Distributed Representations of Words

Neural networks are a powerful modeling tool for many machine learning tasks; NLP is no alternative. But they require continuous inputs in the form of vectors. At first glance, it might be unclear how to represent words, which are atomic units, into continuous vectors. Before we answer this question, let us first discuss why we want to express words as vectors from a linguistic perspective. To perform well on any NLP task, we need to have some notion of similarity (and difference) between words. With words represented as vectors, we can easily encode this ability in the vectors themselves (e.g., using similarity or distance measures such as Cosine or Euclidean).

There are over 500,000[1] words in the Bengali dictionary. Words in a sentence form a context with their surrounding words and give the sentence meaning. We assume that these meanings

[1] https://w.wiki/56H8

can be encoded in a latent space that is substantially smaller than the 500,000 dictionary size. Each dimension of the latent space would encode some specific linguistic information of the words (e.g., part of speech, named entity or not, singular or plural). These representations are called distributed representations of words [75].

The simplest form of word vectors or word embeddings is the one-hot encoding where words are represented by vectors of size equal to the dictionary with all of their entries 0s except for a one at the position corresponding to the index of the word in the dictionary. Unfortunately, this naive approach fails to encode any notion of similarity (the cosine similarity of any two words is 0, by definition). Moreover, this causes a scaling issue and data sparsity with the increase of the effective vocabulary. But this representation provides a means to encode the words into a hidden dimension of a far fewer size. We describe such a model, the continuous bag of words (CBOW) model [73] below:

CBOW assumes the distributional hypothesis [104]: words that are similar in meaning occur in similar contexts. Therefore, the meaning of a word can be inferred from its surrounding words, named its context. Using this hypothesis, CBOW uses large amounts of unlabeled text corpora to learn word embeddings that can capture the linguistic attributes of the words while being reasonably small in the number of dimensions. Let N be the vocabulary size and n be the latent dimension size ($n \quad N$). We create a neural network with one hidden layer having dimension n while both the input and the output have dimension N. The network therefore has two parameter matrices $U \in \mathrm{R}^{n*N}, V \in \mathrm{R}^{N*n}$. Let x^c be our one-hot word vector of interest. We assume it having a context of size m, i.e., $x^{c-m}, \cdots, x^{c-1}, x^{c+1}, x^{c+m}$ be its surrounding words. As the training objective, we use this context to predict x^c using the distributional hypothesis. We describe the process as follows:

1. We generate the one-hot word vectors:

$$x^{c-m}, \cdots, x^{c-1}, x^{c+1}, \cdots, x^{c+m}$$

.

2. We embed the one-hot vectors into the latent space:

$$u^{c-m} = Ux^{c-m}, \cdots, u^{c-1} = Ux^{c-1}, u^{c+1} = Ux^{c+1}, u^{c+m} = Ux^{c+m}$$

.

3. Average the contexts:

$$h = \frac{u^{c-m} + \cdots + u^{c-1} + u^{c+1} + u^{c+m}}{2m}$$

.

4. Project the average to N dimensions again: $\vartheta = Vh$.

5. Covert the vector to a probability distribution: $\hat{y} = \text{softmax}(\vartheta)$.

The model parameters can be learned the same way as discussed in Section 2.2.1. After we get the trained model, the projection of a one-hot word vector into the latent space (Ux^c) represents the distributed representation of the word in a linguistically rich embedding space.

Neural Language Models

Traditional n-gram language models suffer from two major problems as n grows:

1. **Data sparsity**: Since n-gram LMs compute probability estimate for all possible combinations of n-grams, many n-grams may not appear in the training corpus, and therefore, their probabilities remain uncomputable.

2. **Storage Issue**: The number of n-gram probabilities grows proportionally as n increases. Hence the model size grows and causes a problem in storing them.

We describe two neural language models that have been able to tackle these two problems:

Window-based Neural Language Models

Bengio et al. [13] devised the first line of attack against data sparsity by jointly learning distributed represtation of words and their probability distributions (i.e., a language model). At first the words $w_1, \cdots, w_n$ of a sequence S are embedded into a latest space (i.e., word vectors $e_1, \cdots, e_n$) and their representations are concatenated into one single vector $e = [e_1 : \cdots : e_n]$. Then a softmax probability is computed with the concatenated vector

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{We} + \mathbf{b}) \tag{2.11}$$

This represents the probability of the next word of the sequence w_{n+1} conditioned on the previous terms. By eliminating the count-based joint probabilities computation as in the n-gram language models, window-based neural models combat the data sparsity problem.

Recurrent Neural Network-based Language Models

Window-based language models can condition on a finite-size window of previous words as the weight matrix W increases with the context size e. On the other hand, Recurrent Neural Networks (RNNs) [90] are, in theory, capable of conditioning the model on all previous words in the corpus. By treating a sequence of words as time-series data, RNNs keep their model size fixed regardless of the number of previous words. Like time-series models, RNNs feed one word into the model at a time. RNN language models [74] maintain a hidden state h to contain information of all previous words. At each timestep, two inputs, the hidden state of the previous layer, h_{t-1}, and the input at time step t, x_t, are fed into the hidden layer to compute the hidden state of the current layer, which is then used to compute the probability of the next word:

$$h_t = \tanh(W^{(hh)} h_{t-1} + W^{(hx)} x_t) \tag{2.12}$$

$$\hat{y} = \mathrm{softmax}(W^{(s)} h_t) \tag{2.13}$$

Here $W^{(hh)}$, $W^{(hx)}$, $W^{(s)}$ are the model parameters. It is interesting to note that no matter how long the sequence is, the model parameters remain fixed; thereby, the storage issue can be effectively resolved.

Conditional Text Generation and Sequence-to-Sequence Models

Most problems in NLP can broadly be classified into two categories: (1) natural language understanding [120] and (2) natural language generation [36]. Problems like sequence classification and sequence labeling fall into the first category, while problems like machine translation or text summarization fall into the second. Unlike language modeling, which produces probabilities of single words for its given contexts, machine translation or text summarization are somewhat of a special kind of text generation problem known as conditional text generation. We now provide a formal definition of conditional text generation.

Let an input sentence contain m words $x_1, x_2, \cdots, x_n$. Conditional text generation seeks to produce another sequence consisting n words $y_1, y_2, \cdots, y_n$ so that the likelihood of the generated words is maximized according to some objective function. It is to be noted that the output can be of varying length. Mathematically, the objective can be defined as

$$\log p(\boldsymbol{Y} \mid \boldsymbol{X}) \approx \sum_{i=1}^{n} \log p(\boldsymbol{y_{i+1}} \mid \boldsymbol{X}, \boldsymbol{y_1}, \cdots, \boldsymbol{y_{n-1}}) \tag{2.14}$$

Calculating the exact log-likelihood is computationally intractable; hence an approximation is used such that each output token conditions on the context of the previously generated tokens.

A specific type of models, namely sequence-to-sequence (seq2seq) models [115], are popularly used for conditional text generation. Seq2seq models come with an encoder and a decoder. The encoder processes the inputs, and the decoder iteratively generates the output. Generally, both the encoder and decoder are modeled using standard RNNs. The encoder works like a neural language model, taking the encoder hidden state of the previous timestep and word embedding of the current timestep as inputs to produce the current hidden state. The decoder works similarly with a couple of exceptions:

1. Its hidden state is initialized with the final hidden state of the encoder so that the decoder can be informed of the input context produced by the encoder.
2. In each decoding step, the output of the previous timestep is provided as the input embedding since we do not know what input embedding should be given to the decoder during decoding time.

Transformer Language Models

While RNN language models can overcome data sparsity and storage issues, they still suffer from multiple shortcomings, e.g., long-range dependency, vanishing and exploding gradient problems, and sequential training [89]. Many modifications to the training optimization and RNN architecture [44] have been proposed that have successfully addressed the long-range dependency and vanishing and exploding gradient problems. However, the models are sequential and, therefore, require a sequential training routine. This makes it difficult to speed up their training as modern training hardware leverage parallelism to accelerate training workloads. RNNs have thus fallen out of favor, and a new type of model architecture named Transformers [118] has taken their position. Transformers take advantage of a special type of learning mechanism called self-attention [88], which essentially learns embeddings of words as a weighted average of the embeddings of its surrounding words. This mechanism is highly parallelizable and, therefore, speeds up training multiple times. We give a brief overview of the Tranformer network and how it learns word representations.

Let a sequence S have n words $w_1, \cdots, w_n$. They are first embedded into a latent space. Let their embeddings be $e_1, \cdots, e_n$. These embeddings are multiplied by three matrices called the query matrix (W^Q), key matrix (W^K), and the value matrix (W^V):

$$Q = W^Q E, K = W^K E, V = W^V E \tag{2.15}$$

A probability distribution of the attention weights is then learned using the query and keys:

$$P_{qk} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \tag{2.16}$$

Here the probability is divided by $\sqrt{d_k}$, the dimension of the queries, keys, and values so that the dot product values do not get arbitrarily large. Finally, the values are multiplied by their corresponding probabilities and added to obtain the scaled dot product attention values:

$$\text{Attention}(Q, K, V) = P_{qk}V \tag{2.17}$$

Multiple attentions are performed in parallel and they are finally concatanated to obtain the multihead attention:

$$\text{Multihead}(Q, K, V) = \text{concat}(\text{head}_1, \cdots, \text{head}_h)W \tag{2.18}$$

where each head represents a scaled dot product attention. finally these multihead attention representation is passed through a feed-forward network:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.19}$$

The representations after the feed-forward layer can then be used for classification or generation tasks.

Subword Vocabulary

Language models require a predetermined vocabulary that is ID-mapped into embeddings in the embedding layer. Because of the model capacity, the embedding layer has to be kept limited to a fixed size. This, in turn limits the size of the vocabulary. State-of-the-art language models generally use $\sim 30k - 50k$ size vocabulary. However, the vocabulary of a language can reach millions of words. If words cannot be accommodated in the vocabulary of the model, they are mapped to unknown tokens [70] that can cause performance loss of the model. Another option is to use individual letters as the vocabulary [28] (hence, all words would be segmented at the character level), and each letter would have its embedding. This also causes another major problem: the length of a sequence becomes unusually large, and models fail to generalize.

Subword vocabularies [61, 110, 124] are the middle ground between the word-level and character-level vocabularies. As the name suggests, it segments words into multiple subwords, which are generally more than one character long. This is done in a data-driven way: the subword vocabulary is learned using a training corpus. At first, all words are character-segmented, and then the most frequent characters are joined iteratively. This allows frequent words to remain as whole words in the model vocabulary while at the same time segmenting rare words into multiple subwords instead of mapping them to unknown tokens. This method utilizes the model's capacity to the fullest and achieves strong performance gain on many NLP tasks.

Contextualized Word Representations

Word embedding methods like Word2Vec [75] or FastText [14] are generally static, i.e., their representations remain fixed irrespective of their position or context. However, this is not the case for many words, i.e., their meanings can vary with their context. For example, the English word ‘tie’ can have different meanings depending on its use in a sentence. It can mean the outcome of a game (The match was a tie), a piece of garment (I wore a black tie), and a verb to restain something (I tied the box with a ribbon). Static word embedding completely fails to capture these variations in meanings.

Contextualized word representations have been introduced to address this phenomenon. Generally, words are represented using the embedding layer of a neural network. However, the representations of the deeper layers, for instance, the output layer of an RNN at some timestep can also be interpreted as a representation of a word fed at that timestep. In addition to the words’ own static representation, these outputs also become aware of the words previously it has conditioned on by dint of the hidden state it makes use of.

We show an example of a stacked bidirectional RNN [105] that Peters et al. [91] used to learn contextualized word embeddings. The hidden states are generated by feeding the words from both left-to-right and right-to-left so that they can capture both the left-context and the right-context.

$$\overrightarrow{h_t}^{(i)} = \overrightarrow{W}^{(i)}\overrightarrow{h_t}^{(i-1)} + \overrightarrow{V}^{(i)}\overrightarrow{h_{t-1}}^{(i)} + \overrightarrow{b}^{(i)} \tag{2.20}$$

$$\overleftarrow{h_t}^{(i)} = \overleftarrow{W}^{(i)}\overleftarrow{h_t}^{(i-1)} + \overleftarrow{V}^{(i)}\overleftarrow{h_{t-1}}^{(i)} + \overleftarrow{b}^{(i)} \tag{2.21}$$

The input to layer-i is the output of the previous layer $\overrightarrow{h_t}^{(i-1)}$ and $\overleftarrow{h_t}^{(i-1)}$. The final outputs are then concatenated to get the contextualized word representations [$\overrightarrow{h_t}^{(n)}$: $\overleftarrow{h_t}^{(n)}$] word the word at timestep t. The network can be trained using the language modeling objectives. This is why these representations are named “Embeddings from Language Models (ELMo)”.

Pretrained Language Models

NLP has witnessed a sea change with the emergence of pretrained language models. Pretraining neural language models on large amounts of unannotated text corpora using self-supervised objectives has become standard practice nowadays. This pretraining stage allows the language models to learn general linguistic representations [51] that can later be fine-tuned to achieve state-of-the-art results in many NLP tasks. Pretraining is lucrative as it can make use of large text corpora readily available on the web and automatically label them using self-supervised tasks (e.g., autoregressive language modeling [96], masked language modeling [31], masked span generation [97]) without the need for any human annotation. This saves cost and time. Also, pretrained models are general purpose: they can be fine-tuned on any task with minimal supervision, thus alleviating the need for large annotation datasets and compute once pretrained.

Multilingual Language Models

Typically language models are built for a specific language or language pairs. But it is possible to train a language model [25, 26, 31] on hundreds of languages. This makes it possible to train these models on low-resource language that would otherwise have overfitted if trained in isolation. It also lets low-resource languages enjoy the benefits of positive transfer [92] from high-resource languages. Moreover, it reduces the cost of training language-specific models for all the languages of interest. However, multilingual training can be challenging as low-resource languages can become underrepresented during training [123], and consequently, may underperform when trained with other high-resource languages. Therefore, just like class imbalance is addressed during training using oversampling [12, 34], data from low-resource languages are also upsampled to increase their participation during training. We show one such upsampling algorithm by Conneau et al. [26] that has been successfully used in language model pretraining.

Let $L_1, \cdots, L_n$ be n languages present in a training dataset. Let $f_1, \cdots, f_n$ be their respective counts. We define their unnormalized probability as

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_i}; \forall i \in \{1, 2, \cdots, n\} \tag{2.22}$$

We then use an exponent smoothing factor $\alpha \in [0, 1]$ and normalize the probabilities

$$q_i = \frac{p_i^\alpha}{\sum_{j=1}^{n} p_j^\alpha} \forall i \in \{1, 2, \cdots, n\} \tag{2.23}$$

During training, we sample training data using this new probability distribution. A small value of α effectively skews the probability distribution towards the low-resource languages and increases their frequencies during training.

Low-Resource Machine Translation in the Context of Bengali

The first initiative toward machine translation for Bengali dates back to the 90s. Sinha et al. [113] developed ANGLABHARTI, a rule-based translation system from English to multiple Indian languages, including Bengali. Asaduzzaman and Ali, Dasgupta et al. [9, 30] conducted extensive syntactic analyses to write rules for constructing Bengali parse trees and designed algorithms to transfer between Bengali and English parse trees.

Subsequently, Saha et al. [103] reported an example-based machine translation approach for translating news headlines using a knowledge base. Finally, Naskar et al. [83] described a hybrid between rule-based and example-based translation approaches; here, terminals would end at phrases that would then be looked up in the knowledge base.

The improved translation quality of phrase-based statistical machine translation (SMT) [60] and the wide availability of toolkits thereof [58] created an increased interest in SMT for Bengali-English. Moreover, as SMT was more data-driven, specialized techniques were integrated to account for the low amount of parallel data for Bengali-English. Among many, Roy et al. [100] proposed several semi-supervised techniques; Haffari et al. [41] used active learning to improve SMT; Islam et al. [50] used an additional transliteration module to handle OOV words; Banerjee et al. [11] introduced multilingual SMT for Indic languages, including Bengali.

Although NMT is currently being hailed as state-of-the-art, very few works have been done on NMT for the Bengali-English pair. Dandapat and Lewis [29] trained a deployable general domain NMT model for Bengali-English using sentences aligned from comparable corpora. They combated the inadequacy of training examples by data augmentation using back-translation [109]. Hasan et al. and Mumin et al. [42, 77] also showed with limited parallel data available on the web that NMT provided improved translation for Bengali-English pair. Islam et al. [49] proposed a blending algorithm to complement neural machine translation with rule-based machine translation.

Low-resource machine translation systems are generally not fully supervised. Irvine and Callison-Burch [48] used noisy comparable corpora as weak supervision for low-resource machine translation, Gu et al. [39] used small parallel corpora to fine-tune machine translation models trained on other languages, Johnson et al. proposed [52] zero-shot cross-lingual transfer using multilingual translation models, while Lample et al. [63] proposed fully unsupervised approaches.

Multilingual Abstractive Text Summarization

Rush et al. [81, 102] pioneered neural abstractive summarization, using recurrent attentional seq2seq models [10]. See et al. [108] 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. [35] proposed convolutional seq2seq models and applied them to perform abstractive summarization. Narayan et al. [82] extended the work by integrating topic embeddings with the model.

Pretrained language models have recently been successfully applied to abstractive summarization. Liu and Lapata [68] initialized the encoder and Rothe et al. [99] initialized both the encoder and the decoder of a seq2seq model with the pre-trained BERT [31] weights and fine-tuned the models for abstractive summarization. Raffel et al., Yan et al. [95, 97] used fully pre-trained seq2seq models, while Zhang et al. [128] introduced a summarization-specific pretraining objective to achieve state-of-the-art results on multiple datasets.

Most works on abstractive summarization have focused on English, largely due to a lack of benchmark datasets for other languages. Giannakopoulos et al. [37] 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. [18], and Scialom et al. [107] introduced two new datasets for multilingual summarization, but both were limited to less than ten languages. Moreover, samples for different languages were collected from different sources, exposing them to different types of summarization strategies, which raises questions about the uniformity of the summaries.

Chapter 3

Low-Resource Machine Translation

The advancement of deep learning [10, 118, 124] has played an instrumental role in the development of neural machine translation (NMT) models to achieve state-of-the-art results in several language pairs. But a large number of high-quality sentence pairs must be fed into these models to train them effectively [59], and in fact, the lack of such a corpus affects the performance thereof severely. Although there have been efforts to improve machine translation in low-resource contexts, particularly using, for example, comparable corpora [48], small parallel corpora [39] or zero-shot multilingual translation [52], such languages are yet to achieve noteworthy results [57] compared to high-resource ones. Unfortunately, Bengali, the seventh (fifth) most widely spoken language in the world by the number of (native[1]) speakers,[2] has remained a low-resource language. As of now, only a few parallel corpora for the Bengali language are publicly available [116], and those too suffer from poor sentence segmentation, resulting in poor alignments. They also contain much noise, which, in turn, hurts translation quality [54]. No previous work on Bengali-English machine translation addresses any of these issues.

With the above backdrop, in this work, we develop a customized sentence segmenter for the Bengali language while keeping uniformity with the English side segmentation. We experimentally show that better sentence segmentation that maintains homogeneity on both sides results in better alignments. We further empirically show that the choice of sentence aligner plays a significant role in the number of parallel sentences extracted from document pairs. In particular, we study three aligners and show that combining their results, which we name 'Aligner Ensembling,' increases recall. We introduce 'Batch Filtering,' a fast and effective method for filtering out incorrect alignments. Using our new segmenter, aligner ensemble, and batch filter, we collect 2.75 million high-quality parallel sentences from a wide variety of domains, more than 2 million of which were not previously available. Training our corpus on NMT models, we outperform previous approaches to Bengali-English machine translation by more than 9

[1] https://w.wiki/Psq
[2] https://w.wiki/Pss

BLEU [87] points and also show competitive performance with automatic translators. We also prepared a new test corpus containing 1000 pairs made with extensive manual and automated quality checks. Furthermore, we perform an ablation study to validate the soundness of our design choices.

We release all our tools, datasets, and models for public use. This is the first-ever large-scale study on machine translation for Bengali-English pairs, to the best of our knowledge. We believe that the insights brought to light through our work may give new life to Bengali-English MT that has suffered so far for being low in resources. We also believe that our findings will also help design more efficient methods for other low-resource languages.

Sentence Segmentation

Proper sentence segmentation is an essential pre-requisite for sentence aligners to produce coherent alignments. However, segmenting a text into sentences is not a trivial task since the end-of-sentence punctuation marks are ambiguous. For example, in English, the end-of-sentence period, abbreviations, ellipsis, decimal point, etc., use the same symbol (.). Since either side of a document pair can contain Bengali/English/foreign text, we need a sentence segmenter to produce consistent segmentation in a language-independent manner.

Input:

কাজী মুহWদ ওয়াজেদর একমাÑ পুÑ ছিলেন এ. কে. ফজলুক হক। (তার আদি পৈতৃক নিবাস পটুয়াখালী জেলার বাউফল উপজেলায়।)

Expected Output:

1. কাজী মুহWদ ওয়াজেদর একমাÑ পুÑ ছিলেন এ. কে. ফজলুক হক।
2. (তার আদি পৈতক নিবাস পটুয়াখালী জেলার বাউফল উপজেলায়।)

Polyglot Output:

1. কাজী মুহWদ ওয়াজেদর একমাÑ পুÑ ছিলেন এ.
2. কে.
3. ফজলক হক।
4. (তার আদি পৈতক নিবাস পটুয়াখালী জেলার বাউফল উপজেলায়।5.)

Figure 3.1: Erroneous sentence segmentation by Polyglot

Available libraries supporting both Bengali and English segmentation, e.g., Polyglot [4], do not work particularly well for Bengali sentences with abbreviations, which is common in many domains. For instance, Polyglot inaccurately splits the input sentence in Figure 3.1 into three segments, whereas the English side can successfully detect the non-breaking tokens. This corrupts the first alignment and causes the subsequent broken pieces to be aligned with other sentences, creating a chain of incorrect alignments.

SegTok,[3] a rule-based segmentation library, does an excellent job of segmenting English texts. SegTok uses regular expressions to handle many complex cases, e.g., technical texts, URLs, and abbreviations. We extended SegTok's code to have the same functionality for Bengali texts by adding new rules (e.g., quotations, parentheses, bullet points) and abbreviations identified through analyzing both the Bengali and English side of our corpus, side-by-side enhancing SegTok's English segmentation correctness as well. Our segmenter can now address the issues like the example mentioned and provide consistent outputs in a language-agnostic manner.

We compared the performance of our segmenter on different aligners against Polyglot. We found that despite the number of aligned pairs decreasing by 1.37%, the total number of words on both sides increased by 5.39%, making the resulting parallel corpus richer in content. This also bolsters our hypothesis that Polyglot creates unnecessary sentence fragmentation.

ফজলু ল হক ব ে করগ�ে জল র দ্ক্ষিা �লের বধু� �ম স টরিয়য় ১৮৭৩ স ে লর ২৬ অ�র জ�� ক েরন।	Fazlul Huq was born on 26 October 1873 atSaturia, a prosperous village in the South- ern parts of the district of Bakerganj.

Figure 3.2: One-to-one sentence alignment

১৯১৯ স ে ল হক িখলফত আ�নে ে যগদ ন ক েরন, িকX অসহযে ে গর eে �কং ে �ন ে নত ে দরস ে9 তা র মতপথক্য ে দখা িদয়েছিল।	Huq joined the khilafat movement in 1919. But he had difference with the congress leaders on the question of Non-cooperation.

Figure 3.3: One-to-many sentence alignment

১৮৯০ স ে ল ফজলুল হক বরিশ ল িজল �ল ে থকে এ�, ১৮৯২ স ে ল ে eি সডিe কলেজ ে থকে এফ.এ এবং ১৮৯৪ স ে ল িব.এ পরীক্ষ য় উী� হন। ১৮৯৬ স ে ল কলক তা িবি�দ্যা লয় ে থকে িতন গণিত শ ে �এম.এ িড�ল ভ ক েরন।	Fazlul Huq passed the Entrance Examination in 1890 from the Barisal Zilla School, the FA Examination in 1892 and BA Examination in 1894 from the Presidency College, and obtained the MA degree in Mathe-matics in 1896 from the University of Cal- cutta.

Figure 3.4: Many-to-one sentence alignment

Aligner Selection and Ensembling

Aligner Descriptions

Most available resources for building parallel corpora come in the form of parallel documents, which are exact or near-exact translations of one another. Sentence aligners extract parallel sentences from them, which are then used as training examples for MT models. Aligning

[3] https://github.com/fnl/segtok

sentences from noisy comparable corpora is not straightforward. While one might expect that all sentences of a comparable document pair have one-to-one mappings (Figure 3.2), there might also be one-to-many (Figure 3.3) and many-to-one mappings (Figure 3.4). Some sentence may not even have their corresponding pair on the opposite side (i.e., one-to-zero or zero-to-one) mappings. In some rare cases, there might even be many-to-many mappings. Alignment algorithms that extract parallel sequence pairs have to consider these cases.

Abdul et al. [2] conducted a comparative evaluation of five aligners and showed that the choice of aligner had considerable performance gain by the models trained on the resultant bitexts. They identified three aligners with superior performance: Hunalign [117], Gargantua [17], and Bleualign [111]. We briefly describe their working mechanisms:

1. **Hunalign**: Hunalign is a supervised sentence alignment algorithm. It takes two parallel documents as input and uses a bilingual lexicon or dictionary to produce an approximate translation of the source side by simply replacing words that can be found in the dictionary. It then uses dynamic programming to find the best possible alignment by using the lexical overlaps between sequence pairs. In the process, it removes the zero-to-one and one-to-zero alignments.

2. **Gargantua**: Gargantua is a two-stage unsupervised algorithm. It is unsupervised because it takes no dictionary or lexicon as distant or incidental supervision. In its first stage, it uses the length statistics of the documents to compute an approximate alignment. Then, in the second pass, it filters and updates the alignment using dynamic programming.

3. **Bleualign**: The working mechanism is similar to Hunalign. The difference is that instead of a dictionary, the algorithm uses an auxiliary machine translation model to produce an approximate translation, which is then used to determine the alignment. Instead of any length statistics or direct word overlap, Bleualign uses BLEU scores [87] to match potential sentence pairs.

The results from Abdul et al. [2], however, showed performance only in terms of BLEU score, with no indication of any explicit comparison metric between the aligners (e.g., precision, recall). As such, to make an intrinsic evaluation, we sampled 50 documents from four of our sources (detailed in section 3.3.2) with their sentence counts on either side ranging from 20 to 150. We manually aligned sentences from these documents (i.e., the gold alignment) and removed duplicates, resulting in 3,383 unique sentence pairs. We then aligned the documents with the three aligners using our custom segmenter. Table 3.1 shows performance metrics of the aligners.

Aligner Ensembling and Filtering

From the results in Table 3.1, it might seem that Hunalign should be the ideal aligner choice. But upon closer inspection, we found that each aligner could correctly align some pairs that the other two had failed to do. Since we had started from a low-resource setup, it would be in our best interest if we could combine the data extracted by all aligners. As such, we 'ensembled' the results of the aligners as follows. For each combination of the aligners (4 combinations in total; see Table 3.2), we took the union of sentence pairs extracted by each constituent aligner of the said combination for each document. The performance of the aligner ensembles are shown in Table 3.2. We concatenated the first letters of the constituent aligners to name each ensemble (e.g., HGB refers to the combination of all three of them).

Aligner	**Precision**	**Recall**	**F$_1$**
Hunalign	**93.21**	85.82	**89.37**
Gargantua	84.79	69.32	76.28
Bleualign	89.41	**87.35**	88.37

Table 3.1: Performance metrics of aligners in terms of precision, recall and F_1 scores. The most representative metric, F_1 score indicates that Hunalign outperforms the other aligners.

Ensemble	**Precision**	**Recall**	**F$_1$**
HG	83.52	88.00	85.70
GB	81.11	93.20	86.73
BH	**86.16**	94.76	**90.26**
HGB	78.64	**95.13**	86.10

Table 3.2: Performance metrics of ensembles in terms of precision, recall and F_1 scores. Scores indicate that the BH ensemble outperforms the others while being almost identical in performance compared to the standalone aligner Hunalign due to the drop in prescision.

Ensemble	**Precision**	**Recall**	**F$_1$**
L(1.02)	90.86	80.34	85.28
HG+L(0.96)	**94.09**	86.86	90.33
GB+L(0.98)	92.31	91.52	91.91
BH+L(0.96)	91.91	**93.60**	**92.75**
HGB+L(0.98)	91.52	93.23	92.37

Table 3.3: Performance metrics of filtered ensembles in terms of precision, recall and F_1 scores. Scores indicate that the BH ensemble with a filter of margin 0.96 applied outperforms the other standalone aligners, ensembles, and filtereed ensembles.

Table 3.2 shows that BH achieved the best F_1 score among all ensembles, even 0.89% above the best single aligner Hunalign. Ensembling increased the recall of BH by 8.94% compared to Hunalign but also hurt precision severely (by 7.05%) due to the accumulation of incorrect alignments made by each constituent aligner. To mitigate this effect, we used the LASER[4] toolkit to filter out incorrect alignments. LASER, a cross-lingual sentence representation model, uses similarity scores between the embeddings of candidate sentences to perform as both aligner [106] and filter [21]. We used LASER as a filter on top of the ensembles, varied the similarity margin [8] between 0.90 to 1.10 with 0.01 increment, and plotted the performance metrics

[4] https://github.com/facebookresearch/LASER

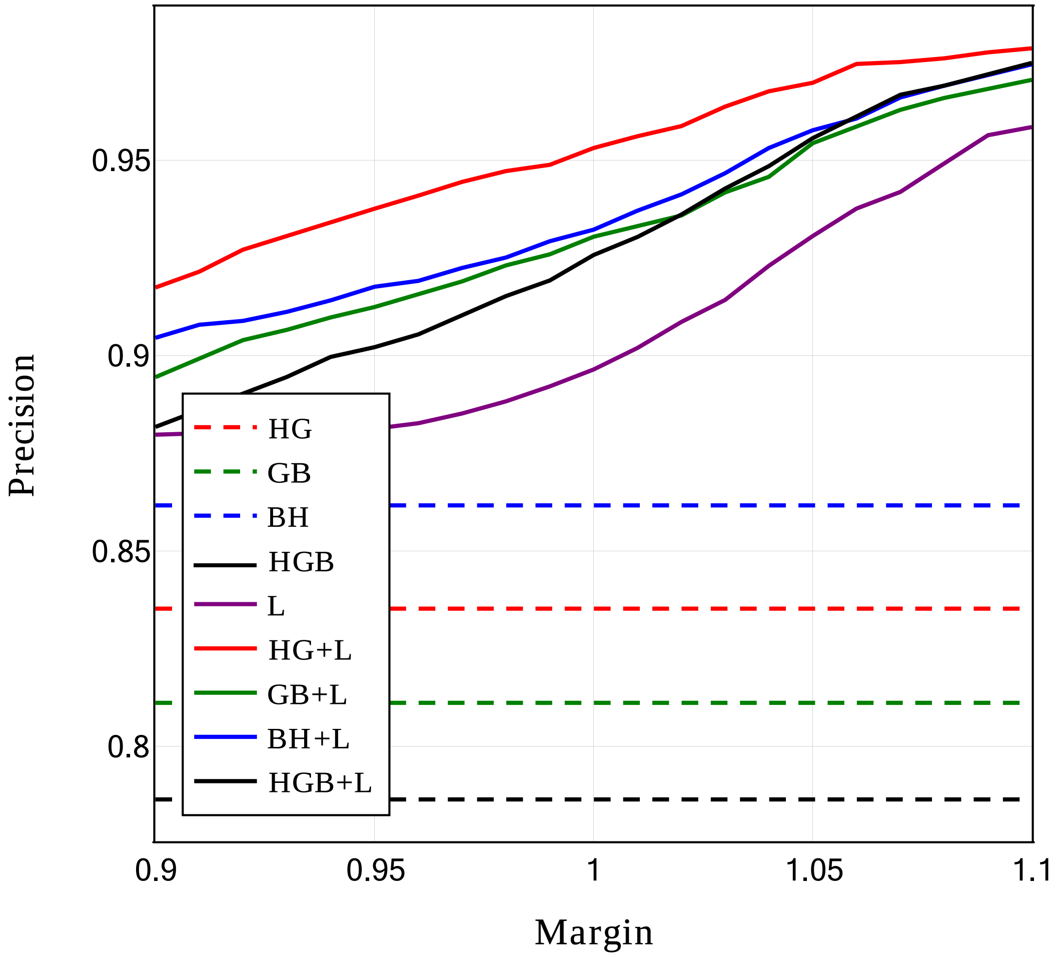


Figure 3.5: Precision vs. Margin

in Figure 3.5, 3.6, 3.7. We also reported the performance of LASER as a standalone aligner (referred to as L in the figure; +L indicates the application of LASER as a filter). The dashed lines indicate ensemble performance without the filter.

As Figure 3.5 indicates, ensembles achieve a significant gain in the precision with the addition of the LASER filter. While recall (Figure 3.6) doesn't face a significant decline at first, it starts to take a deep plunge when the margin exceeds 1.00. We balanced between the two by considering the F_1 score (Figure 3.7). Table 3.3 shows the performance metrics of LASER and all filtered ensembles for which their respective F_1 score is maximized.

Table 3.3 shows that despite being a good filter, LASER as an aligner does not show considerable performance compared to filtered ensembles. The BH ensemble achieves the best F_1 score with

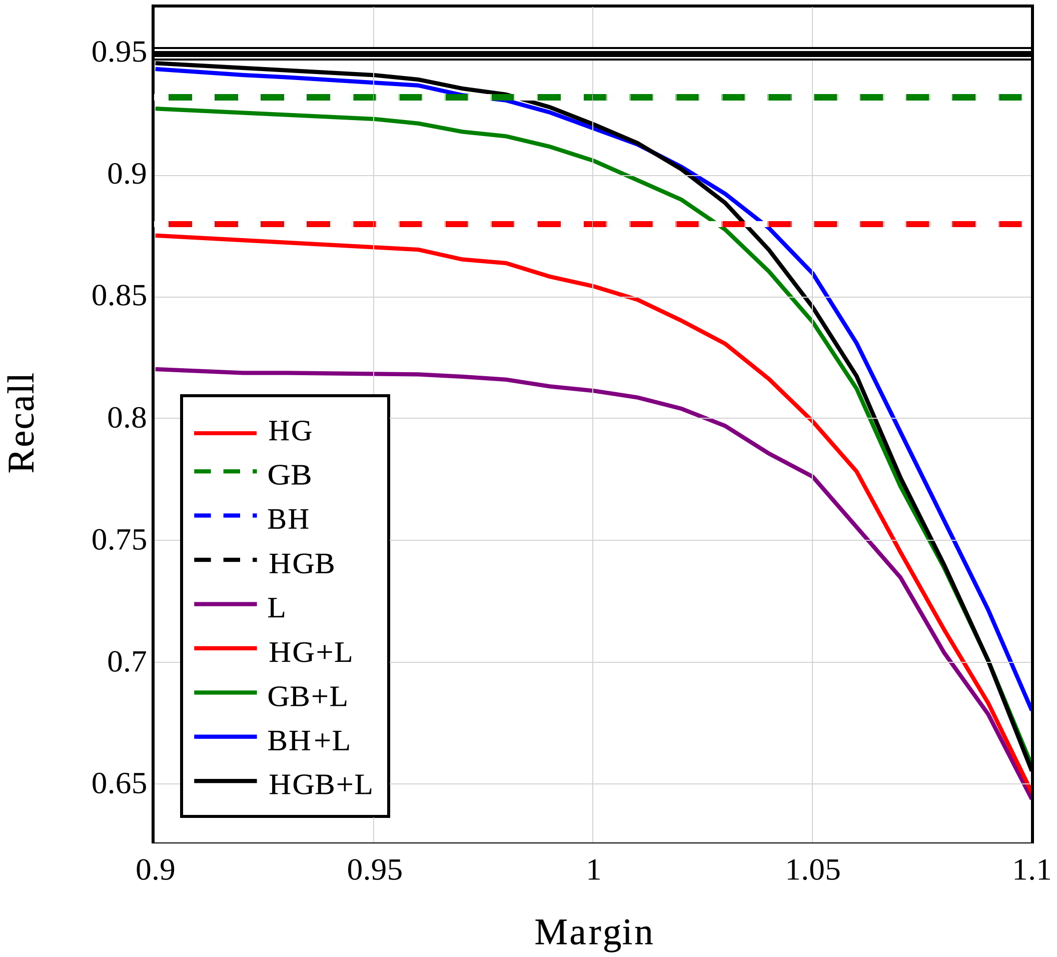


Figure 3.6: Recall vs. Margin

its margin set to 0.96. Its precision increased by 5.75% while trailing a mere 1.16% in recall behind its non-filtered counterpart. Compared to single Hunalign, its recall had a 7.78% gain while lagging in precision by only 1.30%, with an overall F_1 score increase of 3.38%. Thus, we used BH+L(0.96) as our default aligner with the filter margin in all future experiments.

Training Data and Batch Filtering

We categorize our training data into two sections: (1) Sentence-aligned corpora and (2) Document-aligned corpora.

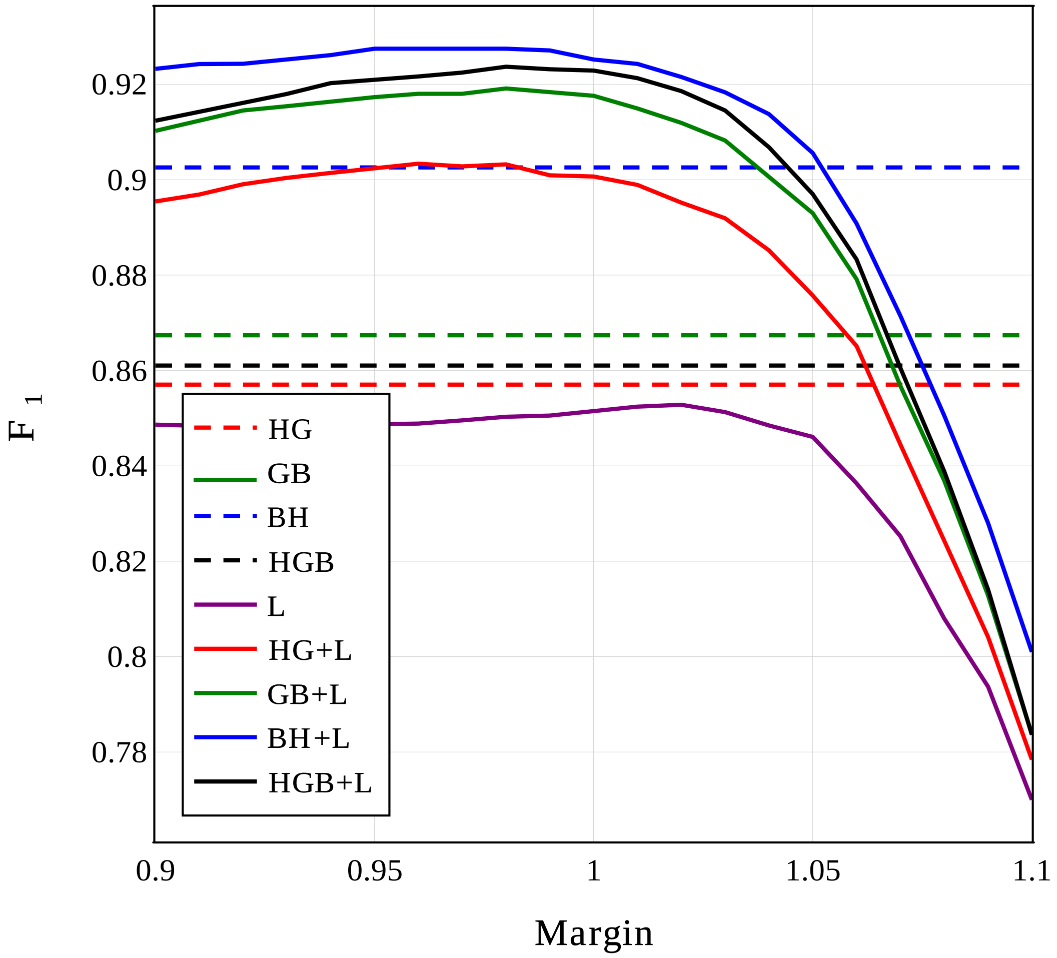


Figure 3.7: F_1 Score vs. Margin

Sentence-aligned Corpora

We used the corpora mentioned below which are aligned by sentences:

1. **Open Subtitles 2018** corpus [67] from OPUS[5] [116]
2. **TED** corpus [19]
3. **SUPara** corpus [79]
4. **Tatoeba** corpus from `tatoeba.org`

[5] `opus.nlpl.eu`

Source	#Pairs	#Tokens(Bn)	#Tokens(En)	#Toks/Sent(Bn)	#Toks/Sent(En)
OpenSubs	365,837	2,454,007	2,902,085	6.71	7.93
TED	15,382	173,149	195,007	11.26	12.68
SUPara	69,533	811,483	996,034	11.67	14.32
Tatoeba	9,293	50,676	57,266	5.45	6.16
Tanzil	5,908	149,933	164,426	25.38	27.83
AMARA	1,166	63,447	47,704	54.41	40.91
SIPC	19,561	240,070	311,816	12.27	15.94
Glosbe	81,699	1,531,136	1,728,394	18.74	21.16
MediaWiki	45,998	3,769,963	4,205,913	81.96	91.44
Gnome	102,078	725,297	669,659	7.11	6.56
KDE	16,992	122,265	115,908	7.20	6.82
Ubuntu	5,251	22,727	22,616	4.33	4.29
Globalvoices	235,106	4,162,896	4,713,335	17.70	20.04
JW	546,766	9,339,929	10,215,160	17.08	18.68
Banglapedia	264,043	3,695,930	4,643,818	14.00	17.59
Books	99,174	1,393,095	1,787,694	14.05	18.03
Laws	28,218	644,384	801,092	22.84	28.39
HRW	2,586	55,469	65,103	21.44	25.17
Dictionary	483,174	700,870	674,285	1.45	1.40
Wiki Sections	350,663	5,199,814	6,397,595	14.83	18.24
Miscellaneous	2,877	21,427	24,813	7.45	8.62
Total	**1,498,338**	**23,847,133**	**27,822,705**	**15.92**	**18.57**

Table 3.4: Summary of the training corpus we curated using the proposed aligner ensembling and batch filtering algorithm. We mention the number of parallel sentences, total number of tokens in Bengali and English, and per sentence average token count in Bengali and English.

5. **Tanzil** corpus from the Tanzil project[6]

6. **AMARA** corpus [1]

7. **SIPC** corpus [94]

8. **Glosbe**[7] online dictionary example sentences

9. **MediaWiki Content Translations**[8]

10. **Gnome, KDE, Ubuntu** localization files

11. **Dictionary** entries from bdword.com

12. **Miscellaneous** examples from english-bangla.com and onubadokderadda.com

[6] tanzil.net/docs/tanzil_project

[7] https://glosbe.com/

[8] https://w.wiki/RZn

Document-aligned Corpora

The corpora below have document-level links from where we sentence-aligned them:

1. **Globalvoices:** Global Voices[9] publishes and translates articles on trending issues and stories from press, social media, and blogs in more than 50 languages. Although OPUS provides a sentence-aligned corpus from Global Voices, we re-extracted sentences using our segmenter and filtered ensemble, resulting in a larger number of pairs compared to OPUS.

2. **JW:** Agic and Vulic [3] introduced JW300, a parallel corpus of over 300 languages crawled from `jw.org`, which also includes Bengali-English. They used Polyglot [4] for sentence segmentation and Yasa [64] for sentence alignment. We randomly sampled 100 sentences from their Bengali-English corpus and found only 23 alignments to be correct. So we crawled the website using their provided instructions and aligned using our segmenter and filtered ensemble. This yielded more than twice the data than theirs.

3. **Banglapedia:** "Banglapedia: the National Encyclopedia of Bangladesh" is the first Bangladeshi encyclopedia. Its online version[10] contains over 5,700 articles in both Bengali and English. We crawled the website to extract the article pairs and aligned sentences with our segmenter and filtered ensemble.

4. **Bengali Translation of Books:** We collected translations of more than 100 books available on the Internet with their genres ranging from classic literature to motivational speeches and aligned them using our segmenter and filtered ensemble.

5. **Bangladesh Law Documents:** The Legislative and Parliamentary Affairs Division of Bangladesh makes all laws available on its website.[11] Some older laws are also available under the "Heidelberg Bangladesh Law Translation Project".[12] Segmenting the laws was not feasible with the aligners in section 3.2.1 as most lines were bullet points terminating in semicolons and treating semicolons as terminals broke down valid sentences. Thus, we made a regex-based segmenter and aligner for these documents. Since most laws were exact translations with equal numbers of bullet points under each section, the deterministic aligner yielded good results.

6. **HRW:** Human Rights Watch[13] investigates and reports on abuses happening in all corners of the world on their website. We crawled the Bengali-English article pairs and aligned them using our segmenter and filtered ensemble.

[9] `https://globalvoices.org/`

[10] `https://www.banglapedia.org/`

[11] `bdlaws.minlaw.gov.bd`

[12] `https://www.sai.uni-heidelberg.de/workgroups/bdlaw/`

[13] `https://www.hrw.org/`

7. **Wiki Sections:** Wikipedia is the largest multilingual resource available on the Internet. But most article pairs are not exact or near-exact translations of one another. However, such a significant source of parallel texts cannot be discarded altogether. Wikimatrix [106] extracted bitexts from Wikipedia for 1620 language pairs, including Bengali-English. But we found them to have issues like foreign texts, incorrect sentence segmentation, alignments, etc. We resorted to the original source and only aligned from sections having high similarities. We translated the Bengali articles into English using an NMT model trained on the rest of our data and compared each section of an article against the sections of its English counterpart. We used SacreBLEU post2018call score as the similarity metric and only picked sections with a score above 20. We then used our filtered ensemble on the resulting matches.

Batch Filtering

LASER uses cosine similarity between candidate sentences as the similarity metric and calculates margin by normalizing over the nearest neighbors of the candidates. Schwenk et al. [106] suggested using a global space, i.e., the complete corpus for neighbor search while aligning, albeit without any indication of what neighborhood to use for filtering. In section 3.2.2, we used local neighborhood on the document level and found satisfactory results. So we tested it with a single aligner, Hunalign,[14] on three large document sources, namely, Globalvoices (GV), JW, and Banglapedia (BP). But the local approach took over a day to filter from about 25k document pairs, the main bottleneck being the loading time for each document. Even with several optimizations, running time did not improve much. The global approach suffered from another issue: memory usage. The datasets were too large to fit into GPU as a whole.[15] Thus, we shifted the neighbor search to CPU, but that again took more than a day to complete. Also, the percentage of filtered pairs was relatively higher than the local neighborhood approach, raising the issue of data scarcity again. So, we sought the following middle-ground between global and local approaches: for each source, we merged all alignments into a single file, shuffled all pairs, split the file into 1k size batches, and then applied LASER locally on each batch, reducing running time to less than two hours.

In Table 3.5, we show the percentage of filtered out pairs from the sources for each neighborhood choice. The global approach lost about twice the data compared to the other two. The 1k batch neighborhood achieved comparable performance with respect to the more fine-grained document-level neighborhood while improving running time by more than ten-folds. Upon further inspection, we found that more than 98.5% pairs from the document-level filter were present in the batched approach. So, in subsequent experiments, we used ‘Batch Filtering’ as the standard. In addition to the document-aligned sources, we also used batch filtering on each

[14] The optimal margin was found to be 0.95 for Hunalign.

[15] We used an RTX 2070 GPU with 8GB VRAM for these experiments.

Source	Document	1k Batch	Global
GV	**4.05**	4.60	8.03
JW	**6.22**	7.06	13.28
BP	**13.01**	14.96	25.65

Table 3.5: Filtered pairs (%) for different neighborhoods. Here, document-level filtering results in the fewest number of sentence pairs filtered out. However, the 1k size batch filter remains very close within a 1-2% margin.

sentence-aligned corpus in section 3.3.1 to remove noise from them. Table 3.4 summarizes our training corpus after the filtering.

We formally describe out aligner ensembling and batch filtering as follows:

Algorithm 1: A pseudocode of the aligner ensembling and batch filtering algorithm.

```
Input: ${S_i, T_j} \forall i \in {1, 2, . . . , n}$: noisy source and target document pairs; Dictionary
       $D$; $M_{mt}$: Auxiliary translation model; $S_g$: Segmenter; $L$: Language-agnostic
       Sentence representation model.

$st_e \leftarrow \varphi$ //ensembled parallel corpus
for ($i \leftarrow 1$ to $n$) do
    ${S_i, T_j} \leftarrow$ {segment$_{S_g}(S_i)$, segment$_{S_g}(T_j)$} //sentence-segmented documents
    $st_h \leftarrow$ Hunalign(${S_i, T_j}, D$) //parallel pairs mined using Hunalign
    $st_b \leftarrow$ Bleualign(${S_i, T_j}, M_{mt}$) //parallel pairs mined using Bleualign
    $st_e \leftarrow st_e \cup st_h \cup st_b$
end
$st_f \leftarrow \varphi$ //filered parallel corpus
$st_e \leftarrow$ split$_{1000}(st_e)$ //ensembled corpus split into batches of size 1000
for ($st \in st_e$) do
    $st \leftarrow$ filter($st, L$) //margin-based filtering on each batch locally
    $st_f \leftarrow st_f \cup st$
end
Output: A parallel corpus $st_f$ curated with a filtered ensemble.
```

Evaluation Data

A major challenge for low-resource languages is the unavailability of reliable evaluation benchmarks that are publicly available. After exhaustive searching, we found two decent test sets and developed one ourselves. They are mentioned below:

1. **SIPC:** Post et al. [94] used crowdsourcing to build a collection of parallel corpora between English and six Indian languages, including Bengali. However, they are not translated by experts and have issues with many sentences (e.g., all capital letters on the English

side, erroneous translations, punctuation incoherence between Bn and En side, presence of foreign texts). They provide four English translations for each Bengali sentence, making it an ideal test-bed for evaluation using multiple references. We only evaluated the performance of Bn→En for this test set.

2. **SUPara-benchmark** [76]: Despite having many spelling errors, incorrect translations, too short (less than 50 characters), and too long sentences (more than 500 characters), due to its balanced nature having sentences from a variety of domains, we used it for our evaluation.

3. **RisingNews:** Since the two test sets mentioned above suffer from many issues, we created our own test set. Risingbd,[16] an online news portal in Bangladesh, publishes professional English translations for many of its articles. We collected about 200 such article pairs and had them aligned by an expert. Next, we had them post-edited by another expert. We then removed, through automatic filtering, pairs that had (1) less than 50 or more than 250 characters on either side, (2) more than 33% transliterations, or (3) more than 50% or more than 5 OOV words [40]. This resulted in 600 validation and 1000 test pairs; we named this test set "**RisingNews**".

Experiments and Results

Pre-processing

Before feeding into the training pipeline, we performed the following pre-processing sequentially:

1. We normalized punctuations and characters with multiple Unicode representations to reduce data sparsity.

2. We removed foreign strings that appear on both sides of a pair, mostly phrases from which both sides of the pair have been translated.

3. We transliterated all dangling English letters and numerals on the Bn side into Bengali, mostly constituting bullet points.

4. Finally, we removed all evaluation pairs from the training data to prevent data leakage.

At this point, a discussion concerning language classification is in order. It is a standard practice to use a language classifier (e.g., FastText [53]) to filter out foreign texts. But when we used it, it classified many valid English sentences as non-English, mainly because they contained named entities transliterated from the Bengali side. Fearing this filtering would hurt the translation of

[16] https://www.risingbd.com/

named entities, we left language classification out altogether. Moreover, most of our sources are bilingual, and we explicitly filtered out sentences with foreign characters so that foreign texts would be minimal.

As for the test sets, we performed minimal pre-processing: we applied character and punctuation normalization. Since SIPC had some sentences that were all capital letters, we lowercased those (and those only).

Comparison with Previous Results

We compared our results with Mumin et al. (2019a) [78], Hasan et al. [42], and Mumin et al. (20219b) [77]. The first work used SMT, while the latter two used NMT models. All of them evaluated on the SUPara-benchmark test set. We used the OpenNMT [55] implementation of big Transformer model [118] with 32k vocabulary on each side learnt by Unigram Language Model with subword regularization[17] [61] and tokenized using SentencePiece [62]. To maintain consistency with previous results, we used lowercased BLEU [87] as the evaluation metric. Comparisons are shown in Table 3.6.

Model	Bn→En	En→Bn
Mumin et al. (2019a) [78]	17.43	15.27
Hasan et al. [42]	19.98	–
Mumin et al. (2029b) [77]	22.68	16.26
Alam et al. (2021) [5]	24.30	–
Ours	**32.10**	**22.02**

Table 3.6: Comparison (BLEU) with previous works on SUPara-benchmark test set (Hasan et al. [42] and Alam et al. [5] did not provide En→Bn scores). Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56].

As evident from the scores in Table 3.6, we outperformed all works by more than **9** BLEU points for Bn→En. Although for En→Bn the difference in improvement (**5.5+**) is not that striking compared to Bn→EN, it is, nevertheless, commendable based on Bengali being a morphologically rich language.

Comparison with Automatic Translators

We compared our models' SacreBLEU[18] post2018call scores with Google Translate and Bing Translator, two most widely used publicly available automatic translators. Results are shown in Table 3.7.

From Table 3.7 we can see that our models have superior results on all test sets when compared to Google and Bing.

[17] l=32, α=0.1

[18] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.4.1 (numrefs.4 for SIPC)

Model /Translator	SUPara Bn→En	SUPara En→Bn	SIPC Bn→En	RisingNews Bn→En	RisingNews En→Bn
Google	29.4	11.1	41.2	33.1	13.3
Bing	24.4	10.7	37.2	28.6	12.1
Ours	**30.7**	**22.0**	**42.7**	**34.5**	**26.2**

Table 3.7: Comparison (SacreBLEU) with automatic translators. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56].

Comparison with Human Performance

Remember that SIPC had four reference English translations for each Bengali sentence. We used the final translation as a baseline human translation and the other three as ground truths (the fourth reference had the best score among all permutations). We evaluated our model's score on the same three references instead of four to make a fair comparison. Human SacreBLEU score was 32.6, while our model scored 38.0, about **5.5** points above human judgment.

Ablation Study of Filtered Ensembles

To validate that our choice of ensemble and filter had direct impact on translation scores, we performed an ablation study. We chose four combinations based on their F_1 scores from section 3.2:

1. Best aligner: **Hunalign**
2. Best aligner with filter: **Hunalign+L(0.95)**
3. Best ensemble: **BH**
4. Best ensemble with filter: **BH+L(0.96)**

We only used data from the parallel documents to ensure apples-to-apples comparison, i.e., Globalvoices, JW, Banglapedia, HRW, Books, and Wiki sections. Table 3.8 shows SacreBLEU scores along with the number of pairs for these combinations. We used the base Transformer model.

Aligner /Ensemble	#Pairs (million)	SUPara Bn→En	SIPC Bn→En
Hunalign	1.35	20.5	33.2
H+L(.95)	1.20	21.0	33.9
BH	**1.64**	21.0	34.0
BH+L(.96)	1.44	**22.1**	**35.7**

Table 3.8: SacreBLEU scores for ablation study. SacreBLEU scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56].

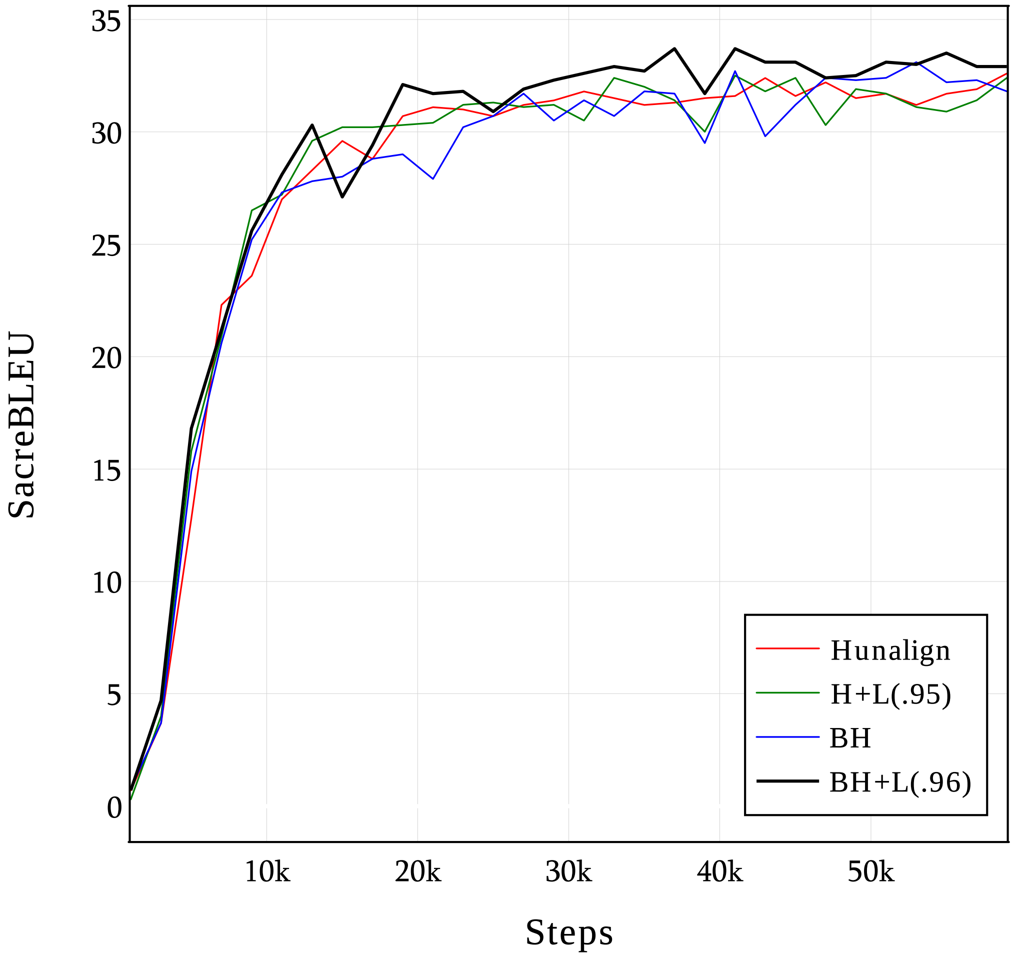


Figure 3.8: SacreBLEU vs Steps on SIPCdev set

BH+L(.96) performed better than others by a noticeable margin, and the single Hunalign performed the poorest. While only having 73% pairs compared to BH, H+L(.95) stood almost on par. Despite the superiority in data count, BH could not perform well enough due to the accumulation of incorrect alignments from its constituent aligners. A clearer picture can be visualized through Figure 3.8. BH+L(.96) mitigated both data shortage and incorrect alignments and formed a clear envelope over the other three, indicating that the filter and the ensemble complemented one another.

Sample Complexity Tests

It is often challenging to annotate training samples in real-world scenarios, especially for low-resource languages like Bangla. So, sample-efficiency [46] is a massive necessity of conditional text generation models. To assess the sample efficiency of aligner ensembling and batch filtering, we limit the number of noisy comparable documents and see how it fares against the standalone alignment methods. We compare it with Hunalign and plot their performances on the SacreBLEU metric for different document counts in Figure 3.10.

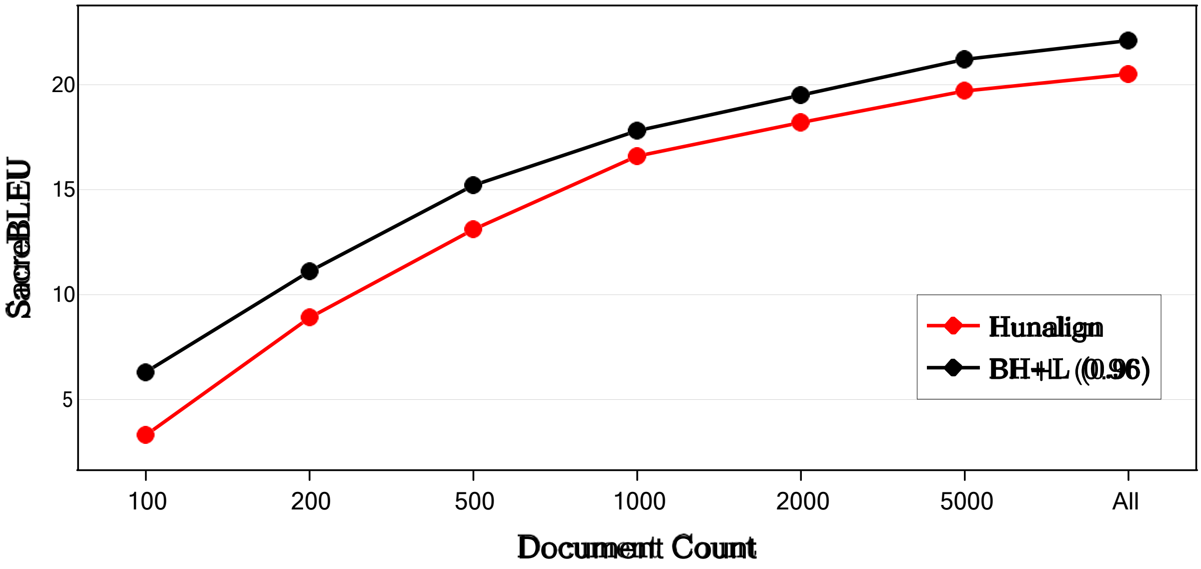


Figure 3.9: SacreBLEU vs Document Count on SUPara test set

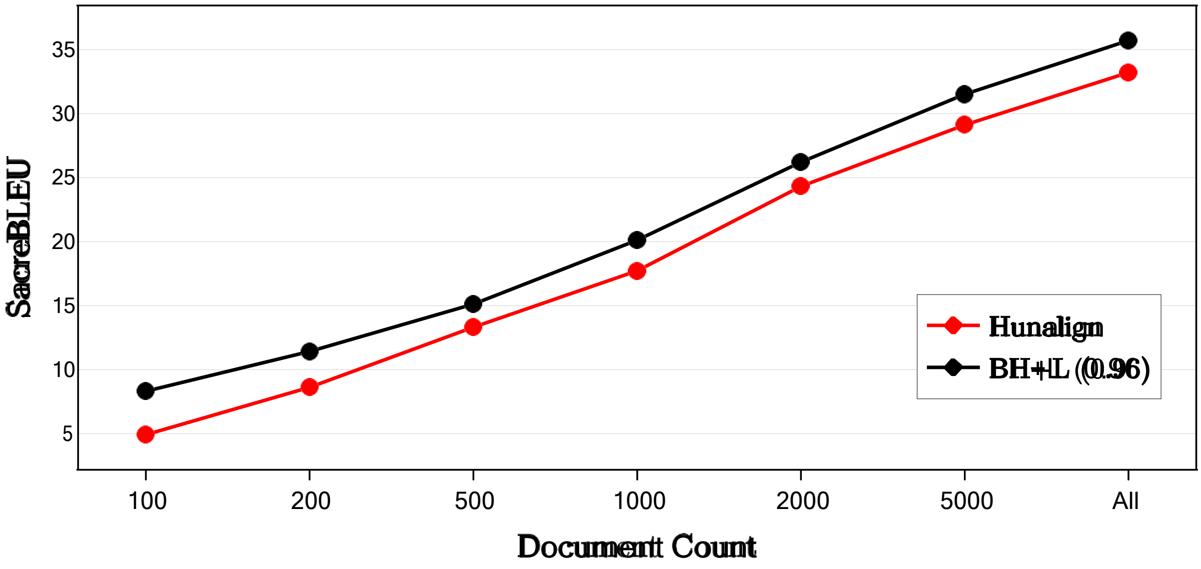


Figure 3.10: SacreBLEU vs Document Count on SIPC test set

Results show that our method consistently outperforms the baseline aligner. And when we have fewer documents ($\leq 500k$), our method has substantially better performance (3-4 SacreBLEU score on SUPara and 2-3 on SIPC with $p < 0.05$) over Hunalign, making it more practically applicable for the resource-scarce machine translation task.

Chapter 4

Multilingual Text Summarization

Automatic text summarization [84] is a fundamental problem in NLP. Given an input text (typically a long document or article), the goal is to 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 [108, 121].

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 [47]. Although abstractive summaries can be more coherent and concise than extractive summaries [24], 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 have been performing better than abstractive ones historically. However, the success of sequence-to-sequence (seq2seq) models [23, 115] over the last decade and the recent advances in Transformer-based models [31, 118] have rejuvenated abstractive text summarization [102,108,128], which had previously received much less attention in comparison to extractive approaches [85]. 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 [38, 43, 82] are available in English only. Though there have been some recent efforts for curating multilingual abstractive summarization datasets [18, 37, 107], 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

Language	#Samples	Language	#Samples
Amharic	7,199	Pidgin[a]	11,510
Arabic	46,897	Portuguese	71,752
Azerbaijani	8,096	Punjabi	10,267
Bengali	10,126	Russian	77,803
Burmese	5,709	Scottish Gaelic	2,313
Chinese (simplified)	46,702	Serbian (Cyrillic)	9,093
Chinese (traditional)	46,713	Serbian (Latin)	9,094
English	329,592	Sinhala	4,249
French	10,869	Somali	7,452
Gujarati	11,397	Spanish	47,636
Hausa	8,022	Swahili	9,872
Hindi	88,472	Tamil	20,276
Igbo	5,227	Telugu	13,025
Indonesian	47,802	Thai	8,268
Japanese	8,891	Tigrinya	6,813
Kirundi	7,182	Turkish	33,970
Korean	5,507	Ukrainian	53,999
Kyrgyz	3,266	Urdu	84,581
Marathi	13,627	Uzbek	5,908
Nepali	7,258	Vietnamese	40,137
Oromo	7,577	Welsh	12,164
Pashto	17,941	Yoruba	7,936
Persian	59,063	**Total**	**1,351,253**

Table 4.1: 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 10,000 training samples to be low-resource, then more than half of the constituent languages in XL-Sum fall into this category.

[a] West African Pidgin English

articles crawled from the British Broadcasting Corporation (BBC)[1] website. We collect 1.35 million professionally annotated article-summary pairs covering 45 languages using a custom crawler. 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 achieved higher than 11 ROUGE-2 scores on the ten languages we benchmark on multilingual summarization, even exceeding 15 ROUGE-2 scores (16.58 being the state-of-the-art for English, obtained by Zhang et al. [128] on XSum [82], a similar dataset) on many of them. In addition, we also conducted experiments on low-resource summarization tasks and showed competitive

[1] https://www.bbc.com/

results, indicating that the dataset can be used even for the low-resource languages individually. In summary, we make the following main contributions:

- We release XL-Sum, a dataset containing 1 million article-summary pairs in 45 languages, 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 diverse 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.

The XL-Sum Dataset

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 4.2: 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.

In this section, we present details of the XL-Sum dataset together with the curation process. Table 4.1 shows the article-summary statistics for all languages in the XL-Sum dataset.

Content Source

BBC publishes news in 43 languages[2] ranging from low-resource languages such as Bengali and Swahili to high-resource ones including English and Russian. Among the 43 languages, Chinese

[2] https://www.bbc.co.uk/ws/languages

and Serbian are exceptional cases: Chinese is published in both simplified and traditional scripts, and Serbian is published in Cyrillic, the official script, and Latin, the colloquial script. We treat them as different languages in this work, totaling a coverage of 45 languages.

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 on 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. Before further processing, we discarded pages with no textual content (mostly pages consisting of multimedia content).

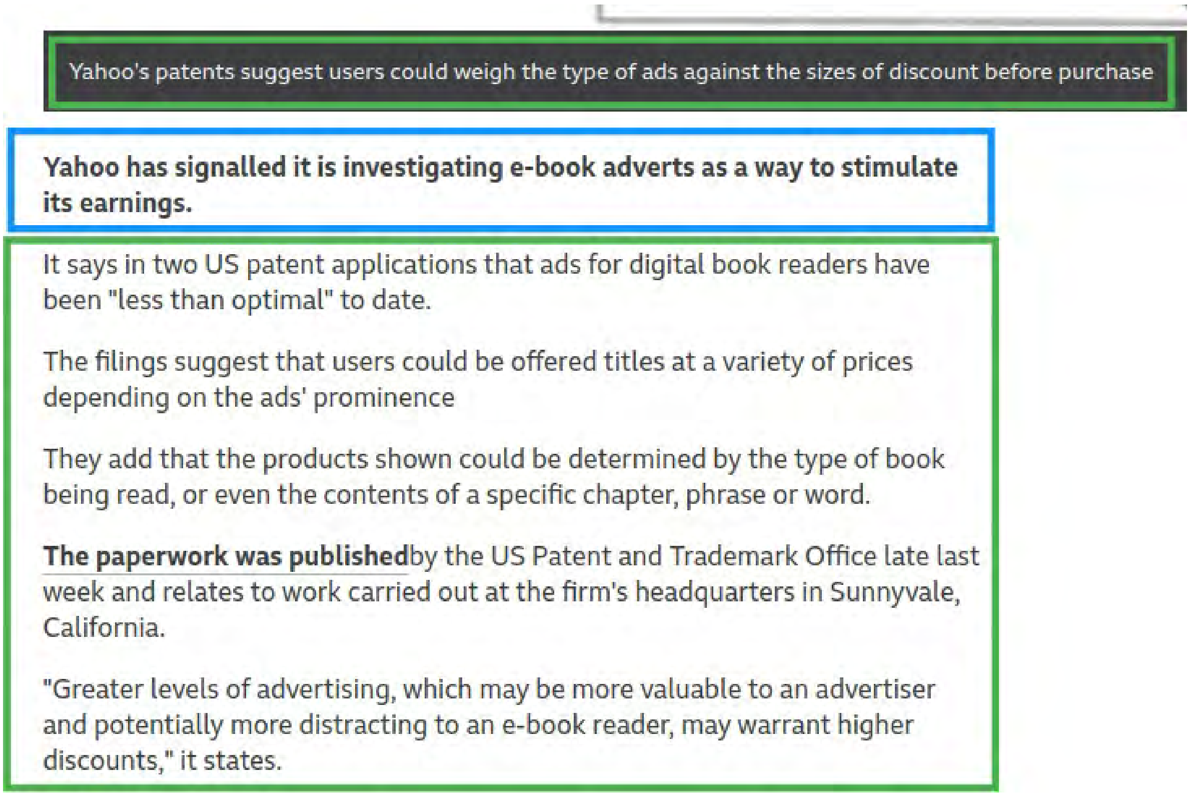
Article-Summary Extraction

The process of automatically collecting summaries of articles differs across different datasets. For example, the CNN/DM dataset [43] merged bullet point highlights provided with the articles as reference summaries. In contrast, the XSum dataset [82] 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 articles' authors to convey the main story within one small paragraph. This contrasts with the headline, which draws viewers' attention to reading the article. (We show an example article-summary pair from BBC English in Table 4.2 and its corresponding HTML page in Figure 4.1.) We designed several 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 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 was discarded. Our strategy of automatic annotation of summaries resembles XSum to some extent. Still, we found the first line to contain



Yahoo's patents suggest users could weigh the type of ads against the sizes of discount before purchase

Yahoo has signalled it is investigating e-book adverts as a way to stimulate its earnings.

It says in two US patent applications that ads for digital book readers have been "less than optimal" to date.

The filings suggest that users could be offered titles at a variety of prices depending on the ads' prominence

They add that the products shown could be determined by the type of book being read, or even the contents of a specific chapter, phrase or word.

The paperwork was publishedby the US Patent and Trademark Office late last week and relates to work carried out at the firm's headquarters in Sunnyvale, California.

"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.

Figure 4.1: A portion of an an example HTML page from BBC English News. The bold text (marked inside the blue box) is used as the summary whilst the the rest of texts (marked inside the green boxes) are used as the corresponding input article.

meta-information in many articles (e.g., author information, date of last modification). As such, we used the bold paragraphs as the summaries instead.

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 [16, 38, 82] 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. [82] 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. [38] introduced *fragments*, which greedily match text spans between the article and the summary, and [16] generalized it to introduce *abstractivity* to measure absractiveness.

Compression: Bommasani et al. [16] proposed *compression* as a metric to quantify conciseness.

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 4.3: 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.

Compression is measured by

$$\mathbf{CMP}(A, S) = 1 - \frac{|S|}{|A|} \tag{4.1}$$

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 et al. [16] 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:

$$\begin{aligned} \mathbf{RED}(S) &= \frac{\sum_{i=1}^{m}(f_i - 1)}{\sum_{i=1}^{m} f_i} \\ &= 1 - \frac{m}{|S| - n + 1} \end{aligned} \tag{4.2}$$

It is preferred for a good summary to have a high novel n-gram ratio, abstractivity, and compression; while having a low redundancy score. We show these metrics in Table 4.3 (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 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, most of the metrics mentioned above indicate that the CNN/Daily Mail dataset is significantly behind XL-Sum and XSum.

Multilingual ROUGE Scoring

RIUGE [66] is the most widely used metric for the automatic evaluation of model-generated summaries. However, it performs some language-specific preprocessing: the official implementation of ROUGE removes non-English characters, tokenizes the texts by words, and then stems the remaining words. Unfortunately, because of the removal of non-English characters, the official implementation is rendered inapplicable to other languages. Therefore, we remove the first processing step, add tokenization support for different languages (especially for languages like Chinese or Japanese, where word boundaries are not determined), and add stemmer support for many languages using open-source libraries [72, 86, 93]. This makes the implementation usable for languages beyond English, and we use that for our model evaluation in the subsequent sections.

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 [118] seq2seq models initialized with pretrained weights from self-supervised training [68, 95, 97, 99, 128] 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 [122]. Among them, we chose to use the mT5 model [125], a multilingual language model pretrained on a large dataset of 101 languages.

We performed summarization experiments in two settings: (i) multilingual and (ii) monolingual. For performance reporting, we used an 80%-10%-10% train-dev-test split for all languages, with a few exceptions. English was split 93%-3.5%-3.5% for the evaluation set, the size resembling that of CNN/DM and XSum; Scottish Gaelic, Kyrgyz, and Sinhala had relatively fewer samples, and their evaluation sets were increased to 500 samples for a more reliable evaluation. The same articles were used to evaluate the two variants of Chinese and Serbian to prevent data leakage in multilingual training.

We tokenized our training samples using the 250k wordpiece [124] 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 [66] scores for automatic evaluation. For inference, we used beam search with beam size four and length penalty of $\alpha = 0.6$ [124].

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 [7] and language model pretraining [27]. However, multilingual training in abstractive summarization has not been a significant focus of the community. As such, this experiment aims to demonstrate that a single model can perform well in 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.

Algorithm 2: A pseudocode of the sampling algorithm for multilingual training.

```
Input: $\mathbf{D}_i\ \forall i \in \{1, 2, \cdots, n\}$: training data having language $L_i$;
    $f_i \leftarrow |\mathbf{D}_i| \forall i \in \{1, 2, \cdots, n\}$; $\mathbf{M}$: Ranomly initialized model
for $i \leftarrow 1$ to $n$ do
    $p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$
    $q_i = \frac{p_i^\alpha}{\sum_{j=1}^{n} p_j^\alpha}$
end
while ($\mathbf{M}$ Not Coverged) do
    Sample $L_i \sim q_i$
    Create batch $b$ from $\mathbf{D}_i$
    Optimize $\mathbf{M}$ using $b$
end
Output: A multilingual model $M$.
```

For this experiment, we followed a similar training strategy as Lample and Conneau [27]: we sampled each batch from a single language containing 256 samples. We used a smoothing factor

(α) of 0.5 so that batches of low-resource languages would be sampled at a higher rate, increasing their frequency during training. We briefly describe the multilingual training algorithm we used in Algorithm 2.

We fine-tuned the mT5 model for 50k steps on a distributed cluster of 8 Nvidia Tesla P100 GPUs for 6 days. We used the Adafactor optimizer [112] 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 4.4.

For comparison, we use a back-translation-based [109] baseline: we translate the source article into English using M2M [32], a massively multilingual machine translation model. Then we pass the translated article through the state-of-the-art English text summarization model, Pegasus, and finally, back-translate the generated summary into our desired language.

Language	Summarization + back-translation			Multilingual Summarization		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
English	**39.79**	**16.58**	**31.70**	37.60	15.15	29.88
Chinese	27.72	7.33	22.19	**39.41**	**17.79**	**33.41**
Hindi	32.51	11.99	26.30	**38.59**	**16.88**	**32.01**
Spanish	30.76	10.72	22.83	**31.51**	**11.88**	**24.07**
French	43.20	14.37	26.60	**35.34**	**16.17**	**28.20**
Arabic	32.12	13.65	28,25	**34.91**	**14.79**	**29.16**
Bengali	22.91	6.17	18.63	**29.57**	**12.11**	**25.13**
Russian	30.24	12.05	24.62	**32.22**	**13.64**	**26.17**
Portuguese	29.87	10.27	22.19	**37.17**	**15.90**	**28.56**
Indonesian	34.37	14.33	28.84	**37.00**	**17.02**	**30.76**

Table 4.4: ROUGE scores for multilingual summarization achieved by the mT5 model when fine-tuned on the XL-Sum training set. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56].

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 several languages, including Bengali.

The mT5-base model achieves an R2-score of 15.18 in the English language. In comparison, the state-of-the-art PEGASUS$_{\text{BASE}}$ model [128] 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. Furthermore, the R-2 scores for other languages are also similar to English, indicating that our dataset can help effectively generate automatic summaries for all languages tested, including those low-resource ones.

Comparison with Monolingual 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, it is still unclear if the low-resource languages can truly reap the full benefit of multilingual training. To confirm this is indeed the case, we performed training on five typologically diverse low-resource languages from Table (Amharic, Azerbaijani, Bengali, Japanese, Swahili) in a monolingual 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 [46]. We show the ROUGE scores of each model in Table 4.5. We use the results from the multilingual models as a baseline.

Language	**Low-resource**			**Multilingual**		
	R-1	R-2	R-L	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	**29.57**	**12.11**	**25.13**
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 4.5: Performance of mT5 model fine-tuned on a low-resource training setup vs multi-lingual setup as mentioned in the previous section. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling [56].

As evident by the results from Table 4.5, the multilingual model outperformed all the models trained on a single language. This confirms the hypothesis that similar languages can indeed have a positive transfer between them [25] 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 [72], and possibly due to its aggressive segmentation, the scores turned out to be higher than in other languages. Similar high results have also been reported while measuring BLEU [87] scores for machine translation evaluation in Japanese [61].

Results in Table 4.5 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 under-served communities.

Chapter 5

Limitations and Ethical Considerations

A significant problem of generative models is the tendency of hallucinations: producing outputs that may not necessarily be faithful to the output. This can be incredibly more severe when the training data itself contains hallucinations. For example, our machine translation training corpus was automatically aligned from noisy comparable corpora. Not all alignments may be correct, i.e., some parallel pairs are not translations of one another. The same can happen for XL-Sum, the text summarization dataset, and there can be additional information in the summaries that may not be present in the corresponding articles. The presence of extra information in the summaries is understandable since professional experts writing these summaries use the information present in the article text and 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 straightforward, making the automatic abstractive summarization task more challenging. This phenomenon may explain why language models fine-tuned on pretrained checkpoints [95, 97, 128] are achieving state-of-the-art results in abstractive summarization, as they can make use of outside information gained from the high volume of texts they were pretrained with.

Dataset and Model Release: The *Copy Right Act, 2000*[1] of Bangladesh allows reproduction and public release of copy-right materials for non-commercial research purposes. We will release the datasets under a non-commercial license as transformative research work. Furthermore, we will release only the data for which we know the distribution will not cause any copyright infringement. The model checkpoints can also be made publicly available under a similar non-commercial license.

Text Content: Text data crawled from the web can often contain offensive or profane texts and personally identifiable information [69]. Models trained on these data may also show different types of implicit bias [15] in them (gender, racial, or religious, to mention a few) that may not be as blatant when looking at the data itself.

[1] http://bdlaws.minlaw.gov.bd/act-details-846.html

We tried to minimize offensive texts by explicitly crawling the sites where such contents would be nominal. However, we cannot guarantee that there is absolutely no objectionable content present and therefore recommend using the datasets and models carefully. Furthermore, we removed the personal information of the content writers by discarding the author fields while collecting the data. Bias is a more complicated issue, and our work does not cover addressing different kinds of biases. Therefore, we give a strong disclaimer to be cautious, especially if the datasets and models are used in production and deployment in the real world.

Chapter 6

Conclusion and Future Works

In this work, we improved two fundamental conditional text generation tasks, machine translation and abstractive text summarization, via low-resource and multilingual modeling techniques. For the machine translation task, we developed a custom sentence segmenter for Bengali, showed that aligner ensembling with batch filtering provides better performance than single sentence aligners, collected a total of 2.75 million high-quality parallel sentences for Bengali-English from multiple sources, trained neural machine translation models that outperformed previous results, and also with Google and Bing translators; thus elevating Bengali from its low-resource status. In the future, we plan to design segmentation-agnostic aligners or aligners that can jointly segment and align sentences. We want to experiment more with the LASER toolkit: we used LASER out-of-the-box, and we want to train it with our data and modify the model architecture to improve it further. LASER fails to identify one-to-many/many-to-one sentence alignments; we want to address this. We would also like to experiment with language-agnostic BERT sentence embeddings [33] embeddings for similarity search. Furthermore, we wish to explore semi-supervised and unsupervised approaches to leverage monolingual data and explore multilingual machine translation for low-resource Indic languages.

For the text summarization task, we present XL-Sum, a large-scale, high-quality multilingual text summarization dataset containing 1.35 million samples across 45 languages collected from a single source, BBC. For many of the languages, XL-Sum provides the first publicly available abstractive summarization dataset and benchmarks. Thorough intrinsic evaluations indicate that the summaries in our dataset are highly abstractive and concise. 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 can also be useful in a low-resource and compute-efficient setting. In the future, we will investigate the use of our dataset for other summarization tasks (e.g., cross-lingual summarization [129]). Finally, 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 for under-served communities with low-resource languages.

Conditional text generation is an umbrella term, and many tasks can be considered part of it. This thesis only addresses text-to-text generation tasks. There are other modes of conditional generation tasks, such as image captioning (image-to-text) [119], speech recognition (speech-to-text) [127], tabular data summarization (data-to-text) [126]. We would also like to explore these domains from a low-resource and multilingual perspective as an extension of our work.

References

[1] ABDELALI, A., GUZMAN, F., SAJJAD, H., AND VOGEL, S. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)* (Reykjavik, Iceland, May 2014), European Language Resources Association (ELRA), pp. 1856–1862.

[2] ABDUL-RAUF, S., FISHEL, M., LAMBERT, P., NOUBOURS, S., AND SENNRICH, R. Extrinsic evaluation of sentence alignment systems. In *Proceedings of the Workshop on Creating Cross-language Resources for Disconnected Languages and Styles* (Istanbul, Turkey, 2012), pp. 6–10.

[3] AGIĆ, Ž., AND VULIĆ, I. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 3204–3210.

[4] AL-RFOU', R., PEROZZI, B., AND SKIENA, S. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (Sofia, Bulgaria, Aug. 2013), Association for Computational Linguistics, pp. 183–192.

[5] ALAM, F., HASAN, A., ALAM, T., KHAN, A., TAJRIN, J., KHAN, N., AND CHOWDHURY, S. A. A review of bangla natural language processing tasks and the utility of transformer models, 2021.

[6] ANDERSON, J. A. *An introduction to neural networks*. MIT press, 1995.

[7] ARIVAZHAGAN, N., BAPNA, A., FIRAT, O., LEPIKHIN, D., JOHNSON, M., KRIKUN, M., CHEN, M. X., CAO, Y., FOSTER, G., CHERRY, C., ET AL. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019* (2019).

[8] ARTETXE, M., AND SCHWENK, H. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the*

Association for Computational Linguistics (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 3197–3203.

[9] ASADUZZAMAN, M., AND ALI, M. M. Morphological analysis of Bangla words for automatic machine translation. In *Proceedings of 6th International Conference on Computers and Information Technology (ICCIT)* (Dhaka, Bangladesh, 2003), pp. 271–276.

[10] BAHDANAU, D., CHO, K., AND BENGIO, Y. 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, 2015).

[11] BANERJEE, T., KUNCHUKUTTAN, A., AND BHATTACHARYA, P. Multilingual Indian language translation system at WAT 2018: Many-to-one phrase-based SMT. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation* (Hong Kong, 1–3 Dec. 2018), Association for Computational Linguistics.

[12] BARUA, S., ISLAM, M. M., YAO, X., AND MURASE, K. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering 26*, 2 (2012), 405–425.

[13] BENGIO, Y., DUCHARME, R., AND VINCENT, P. A neural probabilistic language model. *Advances in Neural Information Processing Systems 13* (2000).

[14] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *Transactions of the association for computational linguistics 5* (2017), 135–146.

[15] BOLUKBASI, T., CHANG, K.-W., ZOU, J., SALIGRAMA, V., AND KALAI, A. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2016), NIPS’16, Curran Associates Inc., p. 4356–4364.

[16] BOMMASANI, R., AND CARDIE, C. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 8075–8096.

[17] BRAUNE, F., AND FRASER, A. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING 2010)* (Beijing, China, Aug. 2010), Association for Computational Linguistics, pp. 81–89.

[18] Cao, Y., Wan, X., Yao, J., and Yu, D. Multisumm: Towards a unified model for multilingual abstractive summarization. In *Proceedings of Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020* (2020), AAAI Press, pp. 11–18.

[19] Cettolo, M., Girardi, C., and Federico, M. Wit3: Web inventory of transcribed and translated talks. In *Proceeding of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)* (Trento, Italy, 2012), European Association for Machine Translation, pp. 261–268.

[20] Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks 20*, 3 (2009), 542–542.

[21] Chaudhary, V., Tang, Y., Guzmán, F., Schwenk, H., and Koehn, P. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)* (Florence, Italy, Aug. 2019), Association for Computational Linguistics, pp. 261–266.

[22] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 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)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1724–1734.

[23] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 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)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1724–1734.

[24] Cohn, T., and Lapata, M. Sentence compression beyond word deletion. In *Proceedingsof the 22nd International Conference on Computational Linguistics (Coling 2008)* (Manchester, UK, Aug. 2008), pp. 137–144.

[25] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 8440–8451.

[26] Conneau, A., and Lample, G. Cross-lingual language model pretraining. *Advances in neural information processing systems 32* (2019).

[27] Conneau, A., and Lample, G. Cross-lingual language model pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)* (Vancouver, Canada, 2019), pp. 7059–7069.

[28] Costa-jussà, M. R., and Fonollosa, J. A. R. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 357–361.

[29] Dandapat, S., and Lewis, W. Training deployable general domain MT for a low resource language pair: English–Bangla. In *Proceeding of the 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)* (Alacant, Spain, 2018), European Association for Machine Translation, pp. 109–117.

[30] Dasgupta, S., Wasif, A., and Azam, S. An optimal way of machine translation from English to Bengali. In *Proceedings of 7th International Conference on Computers and Information Technology (ICCIT)* (Dhaka, Bangladesh, 2004), pp. 648–653.

[31] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)* (Minneapolis, Minnesota, USA, June 2019), Association for Computational Linguistics, pp. 4171–4186.

[32] Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research 22*, 107 (2021), 1–48.

[33] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852* (2020).

[34] Freund, Y., and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences 55*, 1 (1997), 119–139.

[35] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017* (2017), vol. 70, PMLR, pp. 1243–1252.

[36] Gehrmann, S., Adewumi, T., Aggarwal, K., Ammanamanchi, P. S., Aremu, A., Bosselut, A., Chandu, K. R., Clinciu, M.-A., Das, D., Dhole, K.,

Du, W., Durmus, E., Dušek, O., Emezue, C. C., Gangal, V., Garbacea, C., Hashimoto, T., Hou, Y., Jernite, Y., Jhamtani, H., Ji, Y., Jolly, S., Kale, M., Kumar, D., Ladhak, F., Madaan, A., Maddela, M., Mahajan, K., Mahamood, S., Majumder, B. P., Martins, P. H., McMillan-Major, A., Mille, S., van Miltenburg, E., Nadeem, M., Narayan, S., Nikolaev, V., Niyongabo Rubungo, A., Osei, S., Parikh, A., Perez-Beltrachini, L., Rao, N. R., Raunak, V., Rodriguez, J. D., Santhanam, S., Sedoc, J., Sellam, T., Shaikh, S., Shimorina, A., Sobrevilla Cabezudo, M. A., Strobelt, H., Subramani, N., Xu, W., Yang, D., Yerukola, A., and Zhou, J. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)* (Online, Aug. 2021), Association for Computational Linguistics, pp. 96–120.

[37] Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., and Poesio, M. 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* (Prague, Czech Republic, Sept. 2015), Association for Computational Linguistics, pp. 270–274.

[38] Grusky, M., Naaman, M., and Artzi, Y. 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)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 708–719.

[39] Gu, J., Hassan, H., Devlin, J., and Li, V. O. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, USA, June 2018), Association for Computational Linguistics, pp. 344–354.

[40] Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 6098–6111.

[41] Haffari, G., Roy, M., and Sarkar, A. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009*

Annual Conference of the North American Chapter of the Association for Computational Linguistics (Boulder, Colorado, USA, June 2009), Association for Computational Linguistics, pp. 415–423.

[42] HASAN, M. A., ALAM, F., CHOWDHURY, S. A., AND KHAN, N. Neural machine translation for the Bangla-English language pair. In *Proceedings of 22nd International Conference on Computers and Information Technology (ICCIT)* (Dhaka, Bangladesh, 2019), pp. 1–6.

[43] HERMANN, K. M., KOČISKÝ, T., GREFENSTETTE, E., ESPEHOLT, L., KAY, W., SULEYMAN, M., AND BLUNSOM, P. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS 2015)* (Montreal, Canada, 2015), pp. 1693–1701.

[44] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[45] HORNIK, K., STINCHCOMBE, M., AND WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366.

[46] HOWARD, J., AND RUDER, S. 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)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 328–339.

[47] HSU, W.-T., LIN, C.-K., LEE, M.-Y., MIN, K., TANG, J., AND SUN, M. 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)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 132–141.

[48] IRVINE, A., AND CALLISON-BURCH, C. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation* (Sofia, Bulgaria, Aug. 2013), Association for Computational Linguistics, pp. 262–270.

[49] ISLAM, M., ANIK, M., HOQUE, S., ISLAM, A., ET AL. Towards achieving a delicate blending between rule-based translator and neural machine translator. *Neural Computing and Applications 33*, 18 (2021), 12141–12167.

[50] ISLAM, M. Z., TIEDEMANN, J., AND EISELE, A. English to Bangla phrase-based machine translation. In *Proceeding of the 14th Annual Conference of the European*

Association for Machine Translation (EAMT 2010) (St Raphael, France, 2010), European Association for Machine Translation.

[51] JAWAHAR, G., SAGOT, B., AND SEDDAH, D. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics* (2019).

[52] JOHNSON, M., SCHUSTER, M., LE, Q. V., KRIKUN, M., WU, Y., CHEN, Z., THORAT, N., VIÉGAS, F., WATTENBERG, M., CORRADO, G., HUGHES, M., AND DEAN, J. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics 5* (2017), 339–351.

[53] JOULIN, A., GRAVE, E., BOJANOWSKI, P., AND MIKOLOV, T. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (Valencia, Spain, Apr. 2017), Association for Computational Linguistics, pp. 427–431.

[54] KHAYRALLAH, H., AND KOEHN, P. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 74–83.

[55] KLEIN, G., KIM, Y., DENG, Y., SENELLART, J., AND RUSH, A. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations* (Vancouver, Canada, July 2017), Association for Computational Linguistics, pp. 67–72.

[56] KOEHN, P. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 388–395.

[57] KOEHN, P., GUZMÁN, F., CHAUDHARY, V., AND PINO, J. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)* (Florence, Italy, Aug. 2019), Association for Computational Linguistics, pp. 54–72.

[58] KOEHN, P., HOANG, H., BIRCH, A., CALLISON-BURCH, C., FEDERICO, M., BERTOLDI, N., COWAN, B., SHEN, W., MORAN, C., ZENS, R., DYER, C., BOJAR, O., CONSTANTIN, A., AND HERBST, E. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions* (Prague, Czech Republic, June 2007), Association for Computational Linguistics, pp. 177–180.

[59] KOEHN, P., AND KNOWLES, R. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation* (Vancouver, Canada, Aug. 2017), Association for Computational Linguistics, pp. 28–39.

[60] KOEHN, P., OCH, F. J., AND MARCU, D. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (2003), pp. 127–133.

[61] KUDO, T. 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)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 66–75.

[62] KUDO, T., AND RICHARDSON, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Brussels, Belgium, Nov. 2018), Association for Computational Linguistics, pp. 66–71.

[63] LAMPLE, G., OTT, M., CONNEAU, A., DENOYER, L., AND RANZATO, M. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 5039–5049.

[64] LAMRAOUI, F., AND LANGLAIS, P. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment. In *Proceedings of the XIV Machine Translation Summit* (Nice, France, 2013), pp. 77–84.

[65] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[66] LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 74–81.

[67] LISON, P., TIEDEMANN, J., AND KOUYLEKOV, M. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2018), European Language Resources Association (ELRA), pp. 1742–1748.

[68] LIU, Y., AND LAPATA, M. 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) (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3730–3740.

[69] LUCCIONI, A., AND VIVIANO, J. What's in the box? an analysis of undesirable content in the Common Crawl corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Online, Aug. 2021), Association for Computational Linguistics, pp. 182–189.

[70] LUONG, T., SUTSKEVER, I., LE, Q., VINYALS, O., AND ZAREMBA, W. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Beijing, China, July 2015), Association for Computational Linguistics, pp. 11–19.

[71] MAYBURY, M. *Advances in automatic text summarization.* MIT press, 1999.

[72] MCCANN, P. fugashi, a tool for tokenizing Japanese in python. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 44–51.

[73] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013* (2013).

[74] MIKOLOV, T., KARAFIÁT, M., BURGET, L., CERNOCKỲ, J., AND KHUDANPUR, S. Recurrent neural network based language model. In *Interspeech* (2010), vol. 2, Makuhari, pp. 1045–1048.

[75] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., AND DEAN, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems 26* (2013).

[76] MUMIN, M. A. A., SEDDIQUI, M. H., IQBAL, M. Z., AND ISLAM, M. J. SUPara-benchmark: A benchmark dataset for English-Bangla machine translation. In *IEEE Dataport* (2018).

[77] MUMIN, M. A. A., SEDDIQUI, M. H., IQBAL, M. Z., AND ISLAM, M. J. Neural machine translation for low-resource English-Bangla. *Journal of Computer Science 15*, 11 (Nov. 2019), 1627–1637.

[78] MUMIN, M. A. A., SEDDIQUI, M. H., IQBAL, M. Z., AND ISLAM, M. J. shu-torjoma: An English ↔ Bangla statistical machine translation system. *Journal of Computer Science 15*, 7 (Jul. 2019), 1022–1039.

[79] MUMIN, M. A. A., SHOEB, A. A. M., SELIM, M. R., AND IQBAL, M. Z. SUPara: a balanced English-Bengali parallel corpus. *SUST Journal of Science and Technology 16*, 2 (2012), 46–51.

[80] NAIR, V., AND HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In *Icml* (2010).

[81] NALLAPATI, R., ZHOU, B., DOS SANTOS, C., GUÌ‡LÇEHRE, Ç., AND XIANG, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 280–290.

[82] NARAYAN, S., COHEN, S. B., AND LAPATA, M. 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* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 1797–1807.

[83] NASKAR, S. K., AND BANDYOPADHYAY, S. A phrasal EBMT system for translating English to Bengali. In *Proceedings of the Tenth Machine Translation Summit* (Phuket, Thailand, 2005), pp. 372–279.

[84] NENKOVA, A., AND MCKEOWN, K. Automatic summarization. *Foundations and Trends® in Information Retrieval 5*, 2–3 (2011), 103–233.

[85] NENKOVA, A., AND MCKEOWN, K. A survey of text summarization techniques. In *Mining text data*. Springer, 2012, pp. 43–76.

[86] NEUBIG, G., AND MORI, S. Word-based partial annotation for efficient corpus construction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)* (2010).

[87] PAPINENI, K., ROUKOS, S., WARD, T., AND ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (Philadelphia, Pennsylvania, USA, July 2002), Association for Computational Linguistics, pp. 311–318.

[88] PARIKH, A., TÄCKSTRÖM, O., DAS, D., AND USZKOREIT, J. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Austin, Texas, Nov. 2016), Association for Computational Linguistics, pp. 2249–2255.

[89] PASCANU, R., MIKOLOV, T., AND BENGIO, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning* (2013), PMLR, pp. 1310–1318.

[90] PEARLMUTTER, B. A. Learning state space trajectories in recurrent neural networks. *Neural Computation 1*, 2 (1989), 263–269.

[91] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 2227–2237.

[92] PIRES, T., SCHLINGER, E., AND GARRETTE, D. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 4996–5001.

[93] PORTER, M. F. Snowball: A language for stemming algorithms, 2001.

[94] POST, M., CALLISON-BURCH, C., AND OSBORNE, M. Constructing parallel corpora for six Indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation* (Montréal, Canada, June 2012), Association for Computational Linguistics, pp. 401–409.

[95] QI, W., YAN, Y., GONG, Y., LIU, D., DUAN, N., CHEN, J., ZHANG, R., AND ZHOU, M. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), Association for Computational Linguistics, pp. 2401–2410.

[96] RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., AND SUTSKEVER, I. Language models are unsupervised multitask learners. *OpenAI blog 1*, 8 (2019).

[97] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21* (2020), 1–67.

[98] ROTH, D. Incidental supervision: Moving beyond supervised learning. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[99] ROTHE, S., NARAYAN, S., AND SEVERYN, A. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics 8* (2020), 264–280.

[100] ROY, M. A semi-supervised approach to Bengali-English phrase-based statistical machine translation. In *Proceedings of the 22nd Canadian Conference on Artificial Intelligence: Advances in Artificial Intelligence* (Kelowna, Canada, 2009), Springer-Verlag, p. 291–294.

[101] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning representations by back-propagating errors. *nature 323*, 6088 (1986), 533–536.

[102] RUSH, A. M., CHOPRA, S., AND WESTON, J. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 379–389.

[103] SAHA, D., AND BANDYOPADHYAY, S. A semantics-based English-Bengali EBMT system for translating news headlines. In *Proceedings of the Tenth Machine Translation Summit* (Phuket, Thailand, 2005), pp. 125–133.

[104] SAHLGREN, M. The distributional hypothesis. *Italian Journal of Disability Studies 20* (2008), 33–53.

[105] SCHUSTER, M., AND PALIWAL, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing 45*, 11 (1997), 2673–2681.

[106] SCHWENK, H., CHAUDHARY, V., SUN, S., GONG, H., AND GUZMÁN, F. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv:1907.05791* (2019).

[107] SCIALOM, T., DRAY, P.-A., LAMPRIER, S., PIWOWARSKI, B., AND STAIANO, J. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 8051–8067.

[108] SEE, A., LIU, P. J., AND MANNING, C. D. 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)* (Vancouver, Canada, July 2017), Association for Computational Linguistics, pp. 1073–1083.

[109] SENNRICH, R., HADDOW, B., AND BIRCH, A. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 86–96.

[110] SENNRICH, R., HADDOW, B., AND BIRCH, A. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 1715–1725.

[111] SENNRICH, R., AND VOLK, M. MT-based sentence alignment for OCR-generated parallel texts. In *Proceedings of The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)* (Denver, Colorado, USA, 2010).

[112] SHAZEER, N., AND STERN, M. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018* (2018), vol. 80, PMLR, pp. 4603–4611.

[113] SINHA, R., SIVARAMAN, K., AGRAWAL, A., JAIN, R., SRIVASTAVA, R., AND JAIN, A. ANGLABHARTI: a multilingual machine aided translation project on translation from English to Indian languages. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century* (1995), vol. 2, pp. 1609–1614.

[114] SOMERS, H. Machine translation. *The Oxford handbook of translation studies* (1996).

[115] SUTSKEVER, I., VINYALS, O., AND LE, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014)* (Montreal, Canada, 2014), pp. 3104–3112.

[116] TIEDEMANN, J. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)* (Istanbul, Turkey, May 2012), European Language Resources Association (ELRA), pp. 2214–2218.

[117] VARGA, D., HALÁCSY, P., KORNAI, A., NAGY, V., NÉMETH, L., AND TRÓN, V. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP 2005)* (Borovets, Bulgaria, 2005).

[118] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)* (Long Beach, California, USA, 2017), p. 6000–6010.

[119] VINYALS, O., TOSHEV, A., BENGIO, S., AND ERHAN, D. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence 39*, 4 (2016), 652–663.

[120] WANG, A., SINGH, A., MICHAEL, J., HILL, F., LEVY, O., AND BOWMAN, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

[121] WIDYASSARI, A. P., RUSTAD, S., SHIDIK, G. F., NOERSASONGKO, E., SYUKUR, A., AFFANDY, A., ET AL. Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences* (2020).

[122] WOLF, T., DEBUT, L., SANH, V., CHAUMOND, J., DELANGUE, C., MOI, A., CISTAC, P., RAULT, T., LOUF, R., FUNTOWICZ, M., DAVISON, J., SHLEIFER, S., VON PLATEN, P., MA, C., JERNITE, Y., PLU, J., XU, C., LE SCAO, T., GUGGER, S., DRAME, M., LHOEST, Q., AND RUSH, A. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Online, Oct. 2020), Association for Computational Linguistics, pp. 38–45.

[123] WU, S., AND DREDZE, M. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP* (Online, July 2020), Association for Computational Linguistics, pp. 120–130.

[124] WU, Y., SCHUSTER, M., CHEN, Z., LE, Q. V., NOROUZI, M., MACHEREY, W., KRIKUN, M., CAO, Y., GAO, Q., MACHEREY, K., ET AL. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144* (2016).

[125] XUE, L., CONSTANT, N., ROBERTS, A., KALE, M., AL-RFOU, R., SIDDHANT, A., BARUA, A., AND RAFFEL, C. 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* (Online, June 2021), Association for Computational Linguistics, pp. 483–498.

[126] YIN, P., NEUBIG, G., YIH, W.-T., AND RIEDEL, S. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 8413–8426.

[127] YU, D., AND DENG, L. *Automatic speech recognition*, vol. 1. Springer, 2016.

[128] ZHANG, J., ZHAO, Y., SALEH, M., AND LIU, P. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020* (2020), vol. 119, PMLR, pp. 11328–11339.

[129] ZHU, J., WANG, Q., WANG, Y., ZHOU, Y., ZHANG, J., WANG, S., AND ZONG, C. 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)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3054–3064.

Generated using Postgraduate Thesis LaTeX Template, Version 1.03. Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

This thesis was generated on Monday 23rd May, 2022 at 4:58am.