MultiSumm: Towards a Unified Model for Multi-Lingual Abstractive Summarization

Yue Cao,[1,2,3] **Xiaojun Wan,**[1,2,3] **Jin-ge Yao,**[1] **Dian Yu**[4]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Center for Data Science, Peking University
[3]The MOE Key Laboratory of Computational Linguistics, Peking University
[4]Tencent AI Lab
{yuecao, wanxiaojun, yaojinge}@pku.edu.cn, yudian@tencent.com

Abstract

Automatic text summarization aims at producing a shorter version of the input text that conveys the most important information. However, multi-lingual text summarization, where the goal is to process texts in multiple languages and output summaries in the corresponding languages with a single model, has been rarely studied. In this paper, we present MultiSumm, a novel multi-lingual model for abstractive summarization. The MultiSumm model uses the following training regime: (I) multi-lingual learning that contains language model training, auto-encoder training, translation and back-translation training, and (II) joint summary generation training. We conduct experiments on summarization datasets for five rich-resource languages: English, Chinese, French, Spanish, and German, as well as two low-resource languages: Bosnian and Croatian. Experimental results show that our proposed model significantly outperforms a multi-lingual baseline model. Specifically, our model achieves comparable or even better performance than models trained separately on each language. As an additional contribution, we construct the first summarization dataset for Bosnian and Croatian, containing 177,406 and 204,748 samples, respectively.

Introduction

Text summarization has witnessed rapid growth in recent years. There are two primary paradigms for text summarization: extractive and abstractive. Extractive summarization builds summaries by selecting sequences of important sentences and words from the input text. In this paper, we focus on abstractive summarization that can generate a summary that may contain phrases or words that do not appear in the input text.

Variants of sequence-to-sequence model for abstractive summarization have shown to obtain promising results for one single, resource-rich language such as English (Tan, Wan, and Xiao 2017; Lin et al. 2018) and Chinese (Wang et al. 2018; Wei et al. 2019). However, training a monolingual model for each language is neither scalable nor efficient, while for low-resource languages, it is difficult to obtain sufficient training samples for training modern neural models. Therefore, to improve scalability on multiple languages and to improve summarization performance on low-resource languages, it is natural to aim for building a unified multi-lingual model to leverage existing large-scale monolingual summarization corpora in rich-resource languages.

Previous work on multi-lingual text summarization mainly focus on directly mixing training data from different languages, and training with a unified model that does not include modules handling multilingualism (Litvak, Last, and Friedman 2010; Vanetik and Litvak 2015; Litvak et al. 2016; Litvak and Vanetik 2019). These multi-lingual summarization systems are mostly based on traditional machine learning techniques or integer linear programming, which can only handle a small number of training samples. As the number of training samples increases, the time cost of these models becomes unbearable. To the best of our knowledge, there are no deep learning based studies for multi-lingual text summarization.

In this paper, we focus on multi-lingual text summarization. We propose MultiSumm, a unified multi-lingual model to handle multi-lingual text summarization and help improve the summarization performance on those low-resource languages. We study multi-lingual text summarization for five rich-resource languages: English, Chinese, French, Spanish, and German, as well as two low-resource languages: Bosnian and Croatian.

A large number of parallel corpora for English, Chinese, French, Spanish, and German already exist (Graff et al. 2003; Hu, Chen, and Zhu 2015; Mendonça, Graff, and DiPersio 2009a; 2009b), containing millions of text-summary pairs. However, to the best of our knowledge, there is no published summarization dataset available for low-resource languages Bosnian and Croatian. To accomplish the task of text summarization for Bosnian and Croatian, we first create a new abstractive summarization dataset for Bosnian and Croatian, consisting of 177,406 and 204,748 samples, respectively. We give two text-summary examples in Table 1, and the details of the dataset will be described later. Note that the dataset we constructed is relatively smaller than those rich-resource summarization datasets (the ratio of sample size is about 1/20). Due to the lack of training data, traditional deep learning models that contain billions of

An example of Bosnian text-summary pair
Text: Ambasador SAD u Njemačkoj Ričard Grenel ponovo je ukazao na niska ulaganja Berlina u odbranu, rekavši da je to "uvrijedljivo" i ponovio prijetnju da će američke trupe biti povučene iz ove zemlje. (*The US ambassador to Germany, Richard Greer, reiterated Berlin's low investment in defense, saying it was "offensive" and threatening that the US military will withdraw from this country.*) **Summary:** Ambasador SAD prijeti premještanjem američkih trupa iz Njemačke. (*The US ambassafor threatens to withdraw troops from Germany.*)
An example of Croatian text-summary pair
Text: Bivši šampion UFC-a u dvije kategorije i najveća zvijezda MMA, Irac Conor McGregor objavio je da se povlači iz ovog borilačkog sporta u 30. godini, objavili su mediji. (*Former UFC champion in two categories and MMA's biggest star, Irishman Conor McGregor has announced that he is retiring from this sport as the age of 30, media reported.*) **Summary:** Conor McGregor najavio povlačenje. (*Conor McGregor announced his retirement.*)

Table 1: Two examples of Bosnian and Croatian texts with their corresponding summaries. The English translation of the original text is given in brackets.

parameters do not perform well as they could. Thus we aim to leverage the existing summarization corpora for other languages to help the training procedure on low-resource languages.

The training process of our proposed framework can be divided into two stages: (I) multi-lingual learning and (II) joint summary generation training. The multi-lingual learning stage aims at enforcing a shared latent space and helping a model learn the vocabulary and grammar specific to each language, especially for the low-resource languages. In this stage, we train language model, auto-encoder, translation model, and back-translation model for encoders and decoders. In the joint summary generation training stage, we train summarization models for all languages simultaneously.

We conduct experiments on English, Chinese, French, Spanish, German, Bosnian, and Croatian summarization datasets. Experimental results show that our model outperforms the multi-lingual baseline model on all languages. Specifically, our multi-lingual model even surpasses monolingual models on some languages with only $1/7$ parameters of the sum of all monolingual models, and the improvement is significant on the two low-resource languages. In summary, our primary contributions are as follows:

- We propose a new neural model and a new training procedure for multi-lingual text summarization.
- We conduct experiments on summarization datasets in seven languages, and the experimental results show that our model outperforms the multi-lingual baseline model and achieves comparable or better performance than monolingual models.
- We create a new summarization dataset for Bosnian and Croatian, consisting of 177,406 samples and 204,748 samples, respectively.[1]

Related Work

Abstractive Text Summarization

Abstractive text summarization methods typically follow a sequence-to-sequence framework. Rush et al. (2017) first introduce the attention mechanism into the abstractive summarization task. See, Liu, and Manning (2017) propose a copy mechanism that allows the generator to copy words from the source text to alleviate the problem of out-of-vocabulary words. They also propose a coverage mechanism that keeps track of the generated words to discourage repetition. Tan, Wan, and Xiao (2017) introduce a graph-based attention mechanism into the sequence-to-sequence framework to address the saliency factor of text. Celikyilmaz et al. (2018) address the challenges of representing a lengthy text by introducing multiple collaborating agents, each in charge of a subsection of the input text.

Multi-Lingual Text Summarization

Multi-lingual text summarization aims at processing texts in multiple languages and generating summaries in the corresponding languages with a single model. The conference of SIGDIAL-2015 presented a special session for multi-lingual text summarization, named MultiLing 2015 (Giannakopoulos et al. 2015), where most of the recent studies on multi-lingual text summarization were presented (Vanetik and Litvak 2015; Litvak et al. 2016; Litvak and Vanetik 2019). However, the training corpus provided by the session for each language only contains dozens of text-summary pairs, and thus most of the participants use traditional programming-based approaches. Besides, these work achieve multilingualism by copying multiple individual models, each of which trains data written in one language, rather than using one model for multi-lingual summary generation.

To the best of our knowledge, this is the first work to specifically study the multi-lingual text summarization and the first attempt at applying deep learning based methods to multi-lingual text summarization.

Model Architectures

Transformer-Based Summarization Model

Given an input text x consisting of a sequence of M words $\mathbf{x}_1, \ldots, \mathbf{x}_M$, the goal of abstractive text summarization task is to produce a condensed summary y of length $N < M$.

Abstractive text summarization models for a single language mostly adopt sequence-to-sequence architectures, based on LSTM (Sutskever, Vinyals, and Le 2014), CNN (Gehring et al. 2017), transformer (Vaswani et al. 2017), etc.

[1] https://github.com/ycao1996/Multi-Lingual-Summarization

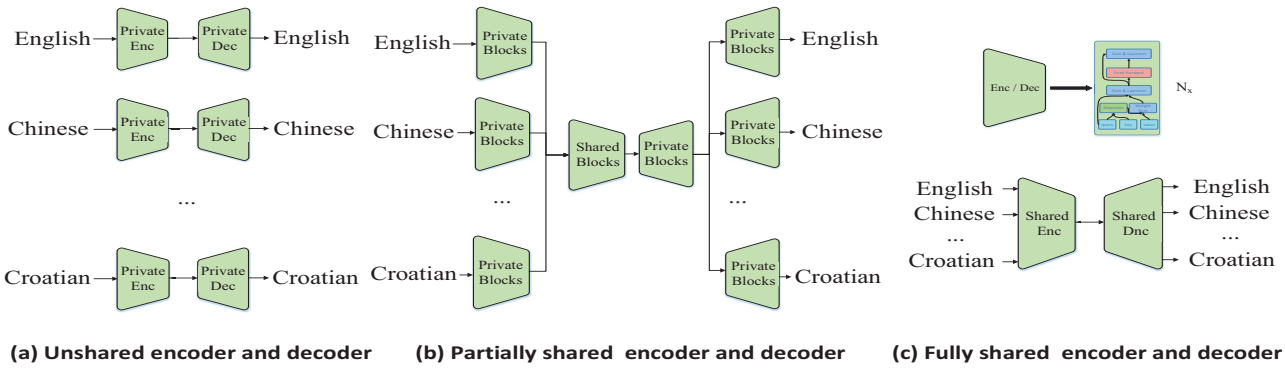

Figure 1: The overall framework of three basic models for multi-lingual summary generation tasks. (a) Unshared model which uses an independent encoder and decoder for each language. (b) Partially shared model which shares the bottom layers of encoders and top layers of decoders across all languages. (c) Fully shared model which uses a fully shared encoder and decoder.

Transformer-based models, which have a potential of modeling very longer-term dependencies, achieve promising results in many sequence-to-sequence tasks. Without loss of generality, we use the transformer (Vaswani et al. 2017) model and extend it to support multi-lingual inputs.

We stack $N = 6$ blocks for both the transformer encoder and decoder. The transformer encoder block contains two sub-blocks: a multi-head self-attention module and a feed-forward module with layer normalization. Additionally, the decoder block has a multi-head cross attention module between the self-attention module and feed-forward module.

We use the same configuration for monolingual models and multi-lingual models.

Sharing Encoders and Decoders

The most straightforward way to support multi-lingual summarization is to train a separate transformer model for each language, which, however, is neither scalable nor efficient. Besides, the possible connections between languages are not exploited.

Instead of training independent models for each language, we build a unified multi-lingual model by partially or fully sharing encoders and decoders across languages. We investigate two ways of sharing encoders and decoders: one is sharing the bottom layers among all encoders and top layers among all decoders, which is depicted in Figure 1(b); the other one is using a shared encoder and a shared decoder for all languages, which is depicted in Figure 1(c). We take the fully shared model as the main model in our experiments and set the partially shared model as a model variation. We also investigate a model variation with unshared encoders and decoders as shown in Figure 1(a). To indicate the target language in the model, we set the first token of the decoder specifies the language the module is operating with.

Subword Embeddings

We process texts in all languages using Byte Pair Encoding (BPE) (Sennrich, Haddow, and Birch 2016). BPE embeddings have been shown to be useful for the alignment of word embedding spaces across languages that share the same alphabet (Conneau et al. 2017; Kim, Gao, and Ney 2019). They can also reduce the vocabulary size and improve the handling of rare words.

We use BPE to process Chinese texts for the following reasons: (1) we tried to use Chinese word segmentation tools to process Chinese texts, but we find the results are not much different from those processed by BPE. (2) Using BPE can greatly reduce the vocabulary size. The Chinese vocabulary size processed by BPE is less than 20K, while the vocabulary size processed by word segmentation tool is at least 100K. (3) Essentially, BPE can be viewed as a word segmentation technique based on the statistical information of the current dataset. It is equivalent to metge the vocabulary with high frequency of co-occurrence on current dataset.

Algorithm 1 Multi-Lingual Training Algorithm for Abstractive Text Summarization

Multi-Lingual Learning:

1: Train language model for all encoders and decoders.
2: Train auto-encoder model for all languages.
3: Train translation and back-translation models.

Joint Summary Generation Training:

4: Train summarization model for all languages simultaneously.

Training Procedure

We summarize our training procedure for multi-lingual text summarization in Algorithm 1. The training procedure contains a **multi-lingual learning** stage and a **joint summary generation training** stage.

Multi-Lingual Learning

Multi-lingual learning plays an essential role in multi-lingual summary generation, especially for those low-resource languages, as it is beneficial for: (1) helping a model learn the vocabulary and grammar specific to each language, (2) enforcing a shared latent space across languages, and (3) leading to learning a better initialization for supervised summarization models.

Our multi-lingual learning stage contains language model training, auto-encoder training, translation training, and back-translation training. The objective is to minimize the sum of losses for these modules:

$$\ell_{total} = \ell_{lan} + \ell_{auto} + \ell_{trans} + \ell_{back} \quad (1)$$

Language Model Training Language model training seeks to learn a probability distribution over sequences of words of all languages. We first pretrain all encoders and decoders using a language model objective. We use a powerful bidirectional language model named masked language model (MLM) (Devlin et al. 2018), which is inspired by the cloze task.

Following Devlin et al. (2018), we randomly sample 15% of the tokens from the input text and replace them with (1) the $[\mathrm{MASK}]$ token 80% of the time, (2) a random token 10% of the time, and (3) keep them unchanged 10% of the time. Then, we use the cross entropy loss to train the language model. Supposing that the prediction of the n-th masked token is $\tilde{x}^{(n)}$, the corresponding ground truth is $x^{(n)}$, and the total number of masked tokens is N_m, the language model loss is calculated as:

$$\ell_{lan}\left(\hat{\mathbf{x}}\right) = \sum_{n=1}^{N_m} \ell_{CE}\left(x^{(n)}, \tilde{x}^{(n)}\right) \quad (2)$$

where ℓ_{CE} represents the cross entropy loss.

Auto-Encoder Training We design an auto-encoder training module to enforce a shared latent space across different languages and help models learn the generation process specific to each language. Besides, auto-encoder is essential for back-translation, especially for those lower-resource languages as there is no available translation corpus for these languages.

In the auto-encoder, an encoder ϕ_E maps the input text to real-vector codes $z_i = \phi_E\left(x_i\right)$, and a decoder ϕ_D attempts to reconstruct the input text from z_i. We train the auto-encoder in a denoising auto-encoder way. Concretely, to prevent the model from simply copying the input, we randomly shuffle the word order of the input text. This can also be seen as a **word-reordering** task. The auto-encoder is trained using cross-entropy loss:

$$\ell_{auto}\left(\hat{\mathbf{x}}\right) = \sum_{n=1}^{N_a} \ell_{CE}\left(x^{(n)}, \phi_D(\phi_E(\hat{x}^{(n)}))\right) \quad (3)$$

where $x^{(n)}$ is the n-th sample, $\hat{x}^{(n)}$ is its permuted version, and N_a is the total number of auto-encoder samples.

We use auto-encoders for all languages at each training step.

Translation and Back-Translation Training After auto-encoder training, we train translation models to further enforce a shared latent space. We assume the availability of a lot of parallel translation samples between two rich-resource languages yet the lack of translation samples for low-resource languages.[2] Recently, back translation has been shown to be useful for unsupervised neural machine translation (Lample et al. 2017; 2018) and low-resource neural machine translation (Gu et al. 2018). We train translation models and back-translation models for rich-resource languages, while we only train back-translation models for low-resource languages.

In translation training, given a source language sentence x_s, we use its corresponding encoder ϕ_{E_s} to encode it into real-vector representations. Then we use the decoder of target language ϕ_{D_t} to read the output of encoders and generate tokens $\tilde{x}_t$ in the target language. We minimize the cross entropy loss between $\tilde{x}_t$ and ground truth x_t:

$$\ell_{trans}\left(\hat{\mathbf{x}}\right) = \sum_{n=1}^{N_t} \ell_{CE}\left(x_t^{(n)}, \tilde{x}_t^{(n)}\right) \quad (4)$$

where N_t is the total number of translation samples.

As for back translation training, given a source language sentence x_s, we first generate a translated sentence in the second language $\tilde{x}_t$. Then we obtain $\tilde{x}_s$ by translating $\tilde{x}_t$ back into the source language. We minimize the cross entropy loss between x_s and $\tilde{x}_s$:

$$\ell_{back}\left(\hat{\mathbf{x}}\right) = \sum_{n=1}^{N_b} \ell_{CE}\left(x_s^{(n)}, \tilde{x}_s^{(n)}\right) \quad (5)$$

where $\tilde{x}_s = \phi_{D_t}(\phi_{E_s}(\tilde{x}_t))$, $\tilde{x}_t = \phi_{D_s}(\phi_{E_t}(x_s))$, and N_b is the number of back translation samples. After each training epoch, we translate texts in one language to all other languages using the current encoder and decoder to form the pseudo-parallel corpus.

Joint Summary Generation Training

We first initialize the model with the parameters learned in the multi-lingual learning stage. These weights will be fine-tuned during the summary generation training stage introduced in this section.

In the summary generation training stage, we train the summarization generation task in a sequence-to-sequence fashion using the transformer architecture. The input is the original text, and the output is a condensed summary that contains the main points of the input text.

Given the input text x, the model generates a summary $\tilde{y}$ that maximizes the output summary probability given the original text: $\tilde{y} = \arg\max_y P(y \mid x)$. We adopt maximum log-likelihood training with cross-entropy loss between generated summary $\tilde{y}$ and ground truth y:

$$\ell_{\text{summ}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} \log P\left(y_t^{(n)} \mid \tilde{y}_{<t}^{(n)}, x^{(n)}\right) \quad (6)$$

where N denotes the number of training samples, and $\tilde{y}_{<t}$ denotes the generated tokens preceding $\tilde{y}_t$. We use the parallel of all languages to train summarization model in this stage.

Variations

To study the importance of different components of our model and training procedure, we evaluate the following variations of our proposed model and training procedure.

[2] There may exist some parallel corpora translated from Bosnian and Croatian to the rich-resource languages, but we do not use them in our experiments.

No language model training Instead of initializing the weights using the language model pretrained weights, we train the auto-encoder model from scratch.

No auto-encoder training As we share the encoder and decoder across languages in our method, training an auto-encoder model can help the model learn the vocabulary and grammar specific to each language and enforce a shared latent space. We argue that auto-encoder is crucial for multilingual summarization, especially for the low-resource languages. To verify this assumption, we include a baseline that removes the auto-encoder training.

No translation or back-translation training To test the importance of translation and back-translation training for text summarization on low-resource languages, we remove the translation training and back-translation training in the multi-lingual learning stage.

Partially shared encoders and decoders We use a partially shared model that shares the bottom layers of encoders and top layers of decoders across languages, as depicted in Figure 1(b).

Experiments

Datasets

Monolingual Dataset We use the Europarl-v5 dataset (KOEHN 2005) for English, German, Spanish, and French. Europarl-v5 contains 1,843,035 English monolingual sentences, 1,772,039 German monolingual sentences, 1,822,021 Spanish monolingual sentences, and 1,855,590 French monolingual sentences.

We use the News-Commentary-v13 dataset (Tiedemann 2012) for Chinese, which contains 361,457 Chinese monolingual sentences.

We use the SETIMES dataset (Tiedemann 2012) for Bosnian and Croatian, which contains 1,228,401 Bosnian monolingual sentences and 1,763,732 Croatian monolingual sentences.

We divide 40% sentences of the monolingual data mentioned above for language model training and 60% for auto-encoder training.

Machine Translation Dataset We use the News-Commentary-v13 dataset (Tiedemann 2012) for translation between English, German, Spanish, French, and Chinese. The number of alignment samples for all language pair varies from 59K to 238K, and the average length for each sentence is 23.8.

Summarization Datasets for Rich-Resource Languages We use the Gigaword dataset for English, French, and Spanish summarization (Graff et al. 2003; Mendonça, Graff, and DiPersio 2009a; 2009b). The number of training samples varies from 1,739K to 3,794K. The average length for the input text is 33.1, and the average length for the summary is 8.6. We use the officially divided training sets, validation sets, and test sets.

We use the LCSTS dataset (Hu, Chen, and Zhu 2015) for Chinese summarization. Following Hu, Chen, and Zhu (2015), we use part I as the training set, part II as the validation set, and samples with 3,4,5 scores in part III as the test set. The number of training pairs, validation pairs, and test pairs are 2,400,591, 10,666, and 725, respectively.

We use the SWISS dataset [3] for German summarization. The total number of German text-summary pairs is 100,000, the average length of input text is 445, and the average length of summary is 22. We use the officially divided training sets, validation sets, and test sets.

We will summarize the detail information for these datasets in the supplemental file.

Dataset Construction for Bosnian and Croatian As there is no existing summarization dataset for the low-resource languages Bosnian and Croatian, we first build a new summarization dataset for the two languages.

The process of constructing our summarization dataset is similar to the process of building datasets such as Gigaword and LCSTS. We crawl the news from a Bosnian news website[4] and a Croatian news website[5], then we use the news description as the original text and use the title as the corresponding summary. A text-summary pair will be filtered out if its title contains dates in the format of "xx.xx.xxxx" because we find these titles only contain useless information such as *"Newsletter, 20.05.2019"*.

After filtering, the total number of Bosnian text-summary pairs is 177,406, and the total number of Croatian text-summary pairs is 204,748. We randomly split 80% of the samples as the training set, 10% of the samples as the validation set, and 10% of the samples as the test set. We show two text-summary examples in Table 1.

Competitive Models

We consider the following three competitive models for comparison.

- **Individual** For each language, we train a monolingual transformer model for text summarization.
- **Individual + pretraining** We train an individual transformer model for text summarization on each language, while the encoder and decoder are first pretrained (i.e., using language model training and auto-encoder training) on monolingual texts in the corresponding language.
- **Multi-baseline** The multi-baseline simply trains summarization generation model for all languages using one model **without** any pretraining process.

Ablation Tests

To study the effectiveness of different components of our model and training procedure, we also test the baselines described in the **Variations** section, including (1) no language model training, (2) no auto-encoder training, (3) no translation or back translation training, and (4) partially shared encocders and decoders. Note that the individual model in the previous section can also be seen as an variant model, using unshared encoders and decoders.

[3] https://www.swisstext.org/shared-task/german-text-summarization-challenge/

[4] https://ba.voanews.com

[5] http://www.federalna.ba

Method	Metric	Rich-Resource					Low-Resource	
		De	En	Es	Fr	Zh	Bs	Hr
Individual	Rouge-1	34.30*	36.54	30.49*	26.70	35.46	13.76*	12.61*
	Rouge-2	14.44*	17.93	11.92	11.72	21.56	4.48*	4.29*
	Rouge-L	29.54*	32.90	25.93	23.41	**33.97**	11.97*	10.96*
Individual + pretraining	Rouge-1	**45.17**	36.59	**31.63**	27.12	**35.74**	18.22*	18.56*
	Rouge-2	**22.14**	**18.07**	**12.37**	11.35	21.57	6.67*	7.06*
	Rouge-L	**40.32**	32.91	**26.38**	**23.63**	33.92	15.75*	16.44*
Multi-baseline	Rouge-1	39.33*	35.06*	29.42*	25.32*	33.52*	20.27*	20.69*
	Rouge-2	18.02*	16.71*	11.80*	11.33*	20.55*	7.84*	8.20
	Rouge-L	34.93*	31.29*	24.53*	22.76*	32.44*	18.74*	19.14
MultiSumm	Rouge-1	43.41	**36.87**	31.18	**27.20**	35.71	**22.47**	**23.04**
	Rouge-2	21.86	17.96	12.24	**11.78**	**21.86**	**8.35**	**8.75**
	Rouge-L	39.77	**33.07**	26.22	23.57	33.61	**19.42**	**19.63**

Table 2: Experimental results of multi-lingual summarization tests. MultiSumm is our proposed model. We highlight in bold the best results (column). Statistically significant improvement ($p < 0.01$) are marked with *.

Cross-Lingual Tests

As a by-product, we investigate the ability of our model to generate cross-lingual summaries. Cross-lingual summarization aims at generating a summary in one language for an input text in a different language.

We regard English as the source language and let our model generate Chinese summaries. To build the test set, we randomly select 100 samples from English Gigaword test sets and manually translate their summaries into Chinese by graduate students who are Chinese-English bilinguals.

Evaluation Metrics

Keeping in line with previous work, we use ROUGE-1 (unigram), ROUGE-2 (bigram), and ROUGE-L (LCS) scores as the evaluation metrics in our experiments.

Implementation Details

We use the Fairseq toolkit (Ott et al. 2019) to implement the architecture. We use the subword-nmt toolkit[6] to process BPE tokens. For multi-lingual models, we learn the BPE merge operations across all languages.

For transformer architectures, the model hidden size, feed-forward hidden size, the number of layers, and the number of heads are 512, 2,048, 6, and 8, respectively. We use the same configuration for all encoders and decoders. In the model variant with partially shared encoders and decoders (Figure 1(b)), we share the bottom four layers of encoders and top four layers of decoders.

For training and inference, the batch size is set to 4,000 for multi-lingual models and 1,000 for individual models. We use Adam optimizer (Kingma and Ba 2014) and use the same parameters and learning rate schedule as previous work (Vaswani et al. 2017). We use warm-up learning rate (Goyal et al. 2017) for the first 4,000 steps, and the initial warm-up learning rate is set to $1e$-7. We use the dropout technique and set the dropout rate to 0.2. We use beam search for inference, and the beam size is set to 5 according to the results on the validation set.

[6]https://github.com/rsennrich/subword-nmt

Results and Analysis

Multi-Lingual Summarization Results

The overall results of multi-lingual text summarization are shown in Table 2. We have the following observations.

First, the pretraining (including language model training and auto-encoder training) on individual model leads to overall gains in Rouge metrics. The improvement is particularly significant on languages with small training data. The individual model with pretraining outperforms the same model without pretraining by a large margin (18.22 v.s 13.76 in Rouge-1 on the Bosnian dataset and 18.56 v.s 12.61 in Rouge-1 on the Croatian dataset). This may be because the model cannot learn the meaning of some words and grammars well from small-scale text summarization datasets, especially those with low frequency, which can be alleviated by the pretraining stage.

Second, the multi-lingual baseline performs worse than most individual models on rich-resource languages (En, Es, Fr, and Zh), with an average drop of about 1-2 points in Rouge-l. The exception is that it performs better on languages with small training data, which may benefit from the shared BPE tokens across languages.

Third, our proposed model MultiSumm outperforms the multi-lingual baseline on all languages and achieves an improvement of 2-4 points in Rouge-1. More importantly, our model gets competitive or even better results than the individual models, with only $1/7$ parameters of the sum of all individual models. The multi-lingual baseline model gets worse results than individual baselines. However, with a pretraining stage, our multi-lingual model can get competitive results compared with the strong individual models.

These results demonstrate the effectiveness of our proposed framework for multi-lingual summarization task.

Ablation Experiment Results

We show the experimental results of ablation test in Table 3.

First, we observe that all modules contribute to the improvement of the performance. After removing any of the

Method	Metric	Rich-Resource					Low-Resource	
		De	En	Es	Fr	Zh	Bs	Hr
MultiSumm	Rouge-1	43.41	**36.87**	**31.18**	**27.20**	**35.71**	22.47	**23.04**
	Rouge-2	**21.86**	**17.96**	**12.24**	**11.78**	**21.86**	**8.35**	**8.75**
	Rouge-L	39.77	**33.07**	**26.22**	**23.57**	**33.61**	**19.42**	**19.63**
w/o language model	Rouge-1	43.03	36.46	30.69	26.92	35.24	22.36	22.47
	Rouge-2	21.19*	17.64	12.00	11.21	21.39	8.26	8.54
	Rouge-L	39.32	32.25	25.87	23.03	33.18	19.12	19.41
w/o auto-encoder	Rouge-1	40.73*	35.19*	29.48*	26.07*	34.16*	21.01*	20.97*
	Rouge-2	18.54*	16.79*	11.24*	11.39	21.09*	7.94*	8.02*
	Rouge-L	36.07*	31.77*	24.86*	22.82*	32.78*	18.90*	19.02*
w/o translation	Rouge-1	41.42*	36.38	30.54*	26.65*	35.08*	21.46*	21.62*
	Rouge-2	19.06*	17.81	11.85	11.64	21.20*	8.12	8.40*
	Rouge-L	37.91*	32.07*	25.46*	22.74*	32.97*	19.04	19.13*
Partially shared encoder and decoder	Rouge-1	**43.60**	36.65	30.88	27.14	35.47	**22.53**	22.78
	Rouge-2	21.85	17.91	12.13	11.63	21.51	8.29	**8.75**
	Rouge-L	**39.91**	33.01	26.15	23.29	33.56	19.30	19.52

Table 3: Experimental results of ablation tests. Statistically significant improvement ($p < 0.01$) are marked with *.

modules, the performance decreases. The pretraining of language model does not have a significant impact on the results. After removing language model pretraining, most of the results get decreased, but not obviously (about 0-0.4 points decrease in Rouge-1). We guess that this is due to the function of language model pretraining can be reflected in the auto-encoder training module. The auto-encoder plays the most important role in the multi-lingual learning stage. After removing the auto-encoder training module, we observe a severe performance degradation. We also find that the model with partially shared encoders and decoders achieves similar results compared with our fully shared model. Considering that the fully shared model has fewer parameters compared with the partially shared variant, it is a more affordable choice to use the fully shared model architecture without much loss in performance.

Cross-Lingual Summarization Tests

We show an example of cross-lingual summarization in Figure 2. We also present the results of two pipeline models: (1) Trans-Summ, which translates the English input to Chinese first and then generates Chinese summaries and (2) Summ-Trans, which generates English summaries first and then translates the English summaries to Chinese.[7]

We can see that all models generate a summary that is close to the reference. However, the Summ-Trans model mistakenly replaces the word *"war"* with *"campaign"*, which may due to the errors made in the translation process. Our model produces the most concise summary, but leaves out some key information such as *"the escalation of the military war"*.[8]

[7]We use the En-Zh translation dataset mentioned above and use the same transformer architecture to train translation model.

[8]Due to space limitations, we present more examples of cross-lingual generation in the supplemental file.

Text: the sri lankan government on wednesday announced the closure of government schools with immediate effect as a military campaign against tamil separatists escalated in the north of the country .

Reference: 随着 军事 战争 升级 ， 斯里兰卡 关闭 了 学校
(*Sri Lankan closed the school with the escalation of the military war*)

Trans-Summ: 斯里兰卡 宣布 关闭 政府 学校
(*Sri Lanka announced the closure of the government school*)

Summ-Trans: 随着 军事 竞选 的 升级 ， 斯里兰卡 政府 关闭 了 学校
(*The Sri Lankan government closed the school with the escalation of the military campaign*)

Ours: 斯里兰卡 政府 关闭 学校
(*Sri Lanka government closed school*)

Figure 2: An example of cross-lingual summary generation.

Conclusions

In this paper, we propose MultiSumm, a unified multi-lingual model for abstractive text summarization. Our training procedure contains two stages: (1) multi-lingual learning that contains language model training, auto-encoder training, translation training, and back-translation training, as well as (2) joint summary generation training. We also construct a new summarization dataset for low-resource languages Bosnian and Croatian, which contains 177,406 and 204,748 samples, respectively.

We conduct experiments on summarization datasets for five rich-resource languages English, Chinese, French, Spanish, and German, as well as for low-resource languages Bosnian and Croatian. Experimental results show that our

proposed framework significantly outperforms the multilingual baseline model. Specifically, our method can even achieve better performance than some individual models with only 1/7 parameters of the sum of all individual models.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), Tencent AI Lab Rhino-Bird Focused Research Program (No.JR201953), and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Celikyilmaz, A.; Bosselut, A.; He, X.; and Choi, Y. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the NAACL-HLT*, 1662–1675.

Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *Proceedings of the ICML*, 1243–1252.

Giannakopoulos, G.; Kubina, J.; Conroy, J.; Steinberger, J.; Favre, B.; Kabadjov, M.; Kruschwitz, U.; and Poesio, M. 2015. Multiling 2015: multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In *Proceedings of the SIGDIAL*, 270–274.

Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Graff, D.; Kong, J.; Chen, K.; and Maeda, K. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia* 4(1):34.

Gu, J.; Hassan, H.; Devlin, J.; and Li, V. O. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

Hu, B.; Chen, Q.; and Zhu, F. 2015. Lcsts: A large scale chinese short text summarization dataset. In *Proceedings of the EMNLP*, 1967–1972.

Kim, Y.; Gao, Y.; and Ney, H. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. *arXiv preprint arXiv:1905.05475*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KOEHN, P. 2005. Europarl: A parallel corpus for statistical machine translation. *Proc. 10th Machine Translation Summit (MT Summit), 2005* 79–86.

Lample, G.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; et al. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the EMNLP*, 5039–5049.

Lin, J.; Xu, S.; Ma, S.; and Su, Q. 2018. Global encoding for abstractive summarization. In *Proceedings of the ACL*, 163–169.

Litvak, M., and Vanetik, N. 2019. *Multilingual Text Analysis*. WORLD SCIENTIFIC.

Litvak, M.; Vanetik, N.; Last, M.; and Churkin, E. 2016. Museec: A multilingual text summarization tool. In *Proceedings of the ACL (System Demonstrations)*, 73–78.

Litvak, M.; Last, M.; and Friedman, M. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the ACL*, 927–936.

Mendonça, A.; Graff, D.; and DiPersio, D. 2009a. French gigaword.

Mendonça, A.; Graff, D.; and DiPersio, D. 2009b. Spanish gigaword second edition.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the NAACL-HLT*, 48–53.

Rush, A. M.; Harvard, S.; Chopra, S.; and Weston, J. 2017. A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the EMNLP*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the ACL*, 1073–1083.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the ACL*, 1715–1725.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tan, J.; Wan, X.; and Xiao, J. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the ACL*, 1171–1181.

Tiedemann, J. 2012. Parallel data, tools and interfaces in opus. In *Eight International Conference on Language Resources and Evaluation*, 2214–2218.

Vanetik, N., and Litvak, M. 2015. Multilingual summarization with polytope model. In *Proceedings of the SIGDIAL*, 227–231.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, L.; Yao, J.; Tao, Y.; Zhong, L.; Liu, W.; and Du, Q. 2018. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In *Proceedings of the IJCAI*, 4453–4460.

Wei, B.; Ren, X.; Zhang, Y.; Cai, X.; Su, Q.; and Sun, X. 2019. Regularizing output distribution of abstractive chinese social media text summarization for improved semantic consistency. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18(3):31.