Bridging the Gap: Cross-Lingual Summarization with Compression Rate

Yu Bai 1, Heyan Huang 1,†, Kai Fan 2,†, Yang Gao 1, Zewen Chi 1, and Boxing Chen 2

1School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
2Machine Intelligence Technology Lab, Alibaba DAMO Academy
{yubai,hhy63,gyang}@bit.edu.cn
{k.fan,boxing.cbx}@alibaba-inc.com

Abstract

Cross-lingual Summarization (CLS), converting a document into a cross-lingual summary, is highly related to Machine Translation (MT) task. However, MT resources are still underutilized for the CLS task. In this paper, we propose a novel task, Cross-lingual Summarization with Compression rate (CSC), to benefit cross-lingual summarization through large-scale MT corpus. Through introducing compression rate, we regard MT task as a special CLS task with the compression rate of 100%. Hence they can be trained as a unified task, sharing knowledge more effectively. Moreover, to bridge these two tasks smoothly, we propose a simple yet effective data augmentation method to produce document-summary pairs with different compression rates. The proposed method not only improves the performance of CLS task, but also provides controllability to generate summaries in desired lengths. Experiments demonstrate that our method outperforms various strong baselines.

1 Introduction

Cross-lingual Summarization (CLS) is a task that converts a document into a summary in another language. Combining the Machine Translation (MT) task and the Monolingual Summarization (MS) task, CLS has attracted interests of many researchers (Zhu et al., 2019, 2020; Cao et al., 2020; Bai et al., 2021). A recent trend is to explore the relationship of these three relevant tasks. Zhu et al. (2019) apply the classic multitask framework of sequence-to-sequence model (Luong et al., 2015), using a unified encoder to share the knowledge between the CLS and the MT. Takase and Okazaki (2020) suggest to use one single Transformer to learn the MT task, the MS task and the CLS task, distinguished by a special token. However, these works simply treat MT as an independent and auxiliary task for cross-lingual summarization. How to better leverage the huge MT corpus up to the hilt still remains a challenge.

To achieve the mentioned goal, we thoroughly probe the relationship between the MT and the CLS. We observe that the MT task can be viewed as a special case of the CLS task via a concept in the summarization domain: compression rate, which refers to the information ratio between the target summary and the original document in the summarization task. If the target summary contains the exact same amount information as the source document, the compression rate becomes 100%. In a cross-lingual scenario, we assume that a CLS task with compression rate of 100%, is the MT task. In practice, it is suggested to adopt the length ratio to define the approximated compression rate in summarization task (Hahn and Mani, 2000; Yeh et al., 2005).

To make our hypothesis concrete, we design a novel task, Cross-lingual Summarization with Compression rate (CSC), integrating the compression rate as an indicator variable into the sequence to sequence model to control how much information should be kept in the target summary. In our proposal, it is unnecessary to differentiate the CLS

---

*Contribution during internship at Machine Intelligence Technology Lab, Alibaba DAMO Academy.
†Corresponding Author
and MT task, in other words, the model merely learns the unified task – CSC. During model optimization, three purposes are expected, 1) the large parallel corpus of MT targets to improve the translation quality, 2) the summarization corpus enables the extraction of key idea, 3) the compression rate controls the length of the output text.

However, relying on the compression variable alone cannot directly achieve the integration of the MT and the CLS, because of the lack of diversity for compression rates in the training data. The compression rate in summarization task usually distributes around 20%, leading to the data scarcity of compression rates between 30% and 90%. If the transition from CLS to MT is not smooth enough, the training of CSC will almost degenerate into the regular multi-task learning. To bridge this gap, we propose a simple but effective method to generate augmented CLS samples with different compression rates. Given a well-annotated CLS data sample, we iteratively delete the less important language units (sentences and words) to shorten the source document, generating a document-summary pair with a larger compression rate. Intuitively, the compression rate transition from the CLS task to the MT task is illustrated in Figure 1.

In order to integrate the continuous compression variable into the summarization model, e.g., Transformer (Vaswani et al., 2017), we discretize the compression rates by grouping them into equally sized bins within $[0, 1]$. We then introduce a set of embedding vectors to represent each bin. Taking as input the compression rate embedding, the quantitative signal enables the model to aware the length of the output. Moreover, adjusting the width of the bins can allow the trade-off between fine-grained controllability and overall generalization.

We conduct experiments in Zh2EnSum CLS dataset (Zhu et al., 2019). Experimental results show that models trained with the CSC task outperforms the strong baselines in various comparable conditions. Additionally, we show the powerful controllability of the proposed CSC models by generating summaries in different lengths as desired, which further brings more practical value to itself.

Our main contributions can be described as follows. (1) We propose Cross-lingual Summarization with Compression rate (CSC), a novel task that benefits cross-lingual summarization through large-scale MT corpus, and modify the traditional transformer architecture for the corresponding task. (2) To bridge the gap between the CLS and the MT task, we propose a data augmentation method to generated CLS samples with different compression rates. (3) The Experiments demonstrate that models trained with CSC can achieve better performance than other tasks.

2 Related Work

2.1 Cross-lingual Summarization

Cross-lingual summarization is a research topic which receives ascending attention of researchers recently. It combines the task of Machine Translation and Text Summarization, providing convenience for people to get information in foreign languages.

Traditional CLS systems produce cross-lingual summaries in a pipeline fashion, employing either first-translate-then-summarize or first-summarize-then-translate pipeline methods (Wan et al., 2010; Wan, 2011; Zhang et al., 2016). Recently, end-to-end methods have been studied deeper. Zhu et al. (2019) propose the first large scale CLS dataset, applying Transformer (Vaswani et al., 2017) to this task. They further use a shared encoder and two separate decoders to conduct the MT task and the CLS task at the same time. Some researchers follow their work and propose various methods (Zhu et al., 2020; Cao et al., 2020; Bai et al., 2021). However, these methods pay less attention to how to fully utilize the enormous MT corpora.

The multi-task model proposed by Takase and Okazaki (2020) is similar to our method, incorporating the MT task, the MS task, and the CLS task simultaneously. Meanwhile, with length information as input, they use a special length-ratio positional encoding to guide the model generating summary of a specified length. However, they do not inspect the relationship between MT and CLS, lacking exploration to bridging the gap between the MT and the CLS.

2.2 Fixed Length Text Summarization

Fixed Length Summarization is a task that demands to generate a target summary in a specific length. Liu et al. (2018) set the desired length as an input to the initial state of the decoder to control the output length. Takase and Okazaki (2019) propose to use length-ratio position embedding to generate summaries in different lengths. Makino et al. (2019) use a length-constraint objective function to optimize the model.
Our work also favors the fashion of length control. The proposed compression rate is used to manipulate the output length of the generated summary. However, our work is different from these works in two main aspects. First, the motivation of our method is to smoothly bridge the MT and the CLS with the proposed data augmentation, not to generate fixed length summaries. Second, by tuning the width of the compression bin, it allows a less fine-grained controllability to achieve better performance in fixed compression rate without target length information.

3 Background

3.1 Cross-lingual Summarization

Formally, we denote two different languages as $A$ and $B$. For the CLS task, it converts a document $D^A = \{x^A_1, x^A_2, \ldots, x^A_n\}$ in language $A$ into a shorter summary $S^B = \{y^B_1, y^B_2, \ldots, y^B_m\}$ in language $B$, where $x^A$ and $y^B$ represent tokens in language $A$ and $B$ while $n$ and $m$ are the lengths of $D^A$ and $S^B$, respectively. Overall, the CLS task $f_{CLS}$ can be denoted as follows:

$$S^B = f_{CLS}(D^A)$$

Note that in a CLS dataset, it is assumed that the monolingual summary $S^A = \{x^A_1, x^A_2, \ldots, x^A_n\}$ is usually available, where the sequence length is denoted as $\hat{n}$. We will leverage these data to construct our augmented data in Section 4.2.

3.2 Machine Translation

Similarly, a Machine Translation (MT) system converts a source text $X^A = \{x^A_1, x^A_2, \ldots, x^A_n\}$ in language $A$ into its translated text $X^B = \{y^B_1, y^B_2, \ldots, y^B_m\}$ in language $B$. $x^A$ and $y^B$ are both tokens while $n$ and $m$ are the lengths of $X^A$ and $Y^B$. The MT task $f_{MT}$ can be described as follows:

$$X^B = f_{MT}(X^A)$$

The scale of MT corpora is typically much larger than that of cross-lingual summarization. However, how to better leverage the MT corpora to benefit the CLS task still remains a challenge.

4 Methods

To better utilize machine translation corpora, we propose a task that incorporates CLS and MT into one unified framework: Cross-Lingual Summarization with Compression Rate (CSC). Meanwhile, we propose to augment the current training data to better help the model produce summaries in different compression rates.

4.1 Cross-Lingual Summarization with Compression Rate

We define the compression rate $\gamma$ as the ratio of the text length between the document and the summary.

$$\gamma = CR(D^A, S^B) = \frac{m}{n}$$

This variable indicates the information compression degree (as known as abstractiveness) of the summarization process. As $\gamma$ becomes smaller, we expect the model to generate more concise summary. It requires the model to be capable of identifying more important information and assemble a new sentence.

The main assumption of this work is that MT task can be seen as a specific case of the CLS when the compression rate is 1. To unify the two tasks, we propose the Cross-lingual Summarization with Compression rate (CSC), producing a cross-lingual summary conditioned on both a source document $D^A$ and a compression rate $\gamma$:

$$S^B = f_{CSC}(D^A, \gamma)$$

However, directly training a CSC model on the existing MT and CLS corpora is nearly the same as the regular multitask learning. Because the training pair with $\gamma = 0.3 \rightarrow 0.9$ is rare, it leads to a huge gap between MT task and CLS task. To circumvent this problem, we design a data augmentation method to close it.

4.2 Data Augmentation

To bridge the Summarization tasks and Machine Translation tasks smoothly, we have to construct summarization samples with different compression rates. Specifically, the augmented summarization dataset should fulfill the following conditions.

Informative The target summary should contain the important information of the source document, enabling the model to learn the summarization task.

Fluency The target summary should be fluent and readable for the model to acquire the generation ability.
Figure 2: An example of our proposed compression rate based data augmentation method.

Algorithm 1 Our proposed data argument method.

Require: \( D^A = \{s_1, s_2, \ldots, s_l\}, S^A, S^B, \gamma, \hat{\gamma} \)

Ensure: \( \hat{\gamma} > \gamma \)

1: \( \hat{D}^A := D^A \)
2: \( \hat{s}_i := \text{rouge} (s_i, S^B), \forall i \in \{1, 2, \ldots, l\} \)
3: \( k := \arg \min_i \{\hat{s}_i\} \)
4: while \( CR(\hat{D}^A \setminus s_k, S^B) < \hat{\gamma} \) do
5: \( \hat{D}^A := \hat{D}^A \setminus s_k \)
6: \( \hat{s}_i := \text{rouge} (s_i, S^B), \forall s_i \in \hat{D}^A \)
7: \( k := \arg \min_i \{\hat{s}_i\} \)
8: end while
9: while \( CR(\hat{D}^A, S^B) < \hat{\gamma} \) do
10: \( w_i := \text{a random word in } s_k \text{ but not in } S^A \)
11: \( \hat{D}^A := \hat{D}^A \setminus w_i \)
12: end while

Uniform The compression rates should be approximately uniformly distributed within the interval \([0, 1]\) to achieve the smooth transition from MT task to CLS task.

To achieve these goals, we propose a simple but effective data augmentation method. It is based on a simple fact: more similar a sentence in the document is to the summary, more important it is. This leads to an intuitive idea: we can keep the target summary unchanged, and gradually delete the less important sentences and words in the document to obtain the relatively larger compression rates.

We leverage the monolingual summaries in CLS dataset to identify the importance of each sentence in the document. We calculate the ROUGE score (Lin, 2004) between \( S^A \) and each sentence \( s_i \) in \( D^A \). Then, we iteratively delete the least important sentence in \( D^A \) to construct \( \hat{D}^A \). The delete process continues until the compression rate of \( \hat{D}^A \) and \( S^B \) increases and reaches the desired \( \hat{\gamma} \). However, because the sentence-level deletion is too coarse, the compression rate of the obtained document-summary pair could deviate from \( \hat{\gamma} \) a lot. So we restore the last deleted sentence such that the compression rate is still lower than \( \hat{\gamma} \). Words in this sentence but not in the monolingual summary \( S^A \) will be deleted randomly until a partially noisy document \( \hat{D}^A \) with \( \hat{\gamma} \) is acquired. Details of this process can be seen in Algorithm 1.

We augment each sample with a sequence of ascending \( \hat{\gamma} \)'s larger than its original \( \gamma \). Concretely, the construction of the new compression rates can be characterized as injecting random perturbations to an arithmetic progression,

\[ \{\hat{\gamma} \leq 1|\hat{\gamma} = \gamma + i \ast 0.1 \ast U(0, 1), i = 1, 2, \ldots, 10\} \]

4.3 Model Architecture

Transformer (Vaswani et al., 2017) is a widely-used architecture in Natural Language Processing. It consists of multiple stacked encoder and decoder Transformer layers. The encoder and decoder layers both have a self-attention block and a feed-forward block. Besides, the decoder layer also has an additional encoder-decoder attention block to acquire information from the source side. The multi-head attention is applied for all the attention modules. Transformer has been applied to Cross-lingual Summarization in previous work (Zhu et al., 2019; Ladhak et al., 2020). In this paper, we modify the architecture to incorporate the compression rate to the summarization model.

In order to take as input the continuous compression rate \( \gamma \), we apply the quantization trick
to embed it into a numerical vector. We first divide \((0, 1]\) into equally sized bins with the width \(\delta\). Each \(\gamma\) is grouped into the bin if it falls into the corresponding interval. Therefore, we can use a set of learned embedding vectors to represent each bin. The vector is added to the embedding of each token to make the model aware the length of target summary, constraining the compression rate in a specific range. Note that the token embedding of both the encoder and the decoder is modified. Details are illustrated in Figure 3.

Occasionally the compression rate of a few MT examples is \(\geq 1\), but we simply set \(\gamma = 1\) for convenience. During training, we can feed the model with the compression rate of the training pair. However, during inference, we cannot derive the exact compression rate from the source alone. As a compromise, we compute the average compression rate of the CLS training data, and feed it to the model for inference. For a more practical scenario, we feed the oracle (or desired) compression rate into the model to observe the upper bound of the performance. We find that the selection of bin width \(\delta\) is crucial to the result, and will analyze this phenomenon in Section 6.

5 Experimental Settings

5.1 Datasets
We conduct our experiments on a CLS dataset: Zh2EnSum (Zhu et al., 2019). This dataset is converted from the Chinese summarization dataset LCSTS (Hu et al., 2015) using a round-trip strategy. Zh2EnSum contains 1,693,713 Chinese-to-English training samples, 3,000 validation samples, and 3,000 testing samples. All the training samples contain a source document, a monolingual summary, and a cross-lingual summary. The augmented data contains 16,188,500 training samples. For Chinese-to-English translation data, we use WMT17 Zh2En dataset which contains 20,616,495 training samples, 2,002 validation samples, 2,001 test samples.

5.2 Baselines and Variations of CSC
We compare our proposed models with the following baselines:

**TETran** is the pipeline model which translates the document first and then summarize it.

**TLTran** is the pipeline model which first summarizes a document into a summary and then translates the summary.

**NCLS** is the vanilla baseline model which use the Transformer model to directly conduct CLS task without any extra data. We also use mBART to initialize the model to acquire a stronger baseline.

**NCLS+MT** is the Transformer-based multi-task framework which conducts CLS task and MT task simultaneously. To investigate the fully potential of it, we experiment with its several variations:

- **Share Decoder (SD)**: the decoder is shared for both tasks, sharing the knowledge of how to generate text in target language.
- **Share Encoder (SE)**: the encoder is shared for both tasks, sharing the knowledge of how to get a better representation of source language.
- **Share All (SA)**: all the parameters are shared for these tasks. We add a task-specific token before source text to distinguish different tasks.

Since models assisted by Monolingual Summarization (MS) corpora are empirically worse than those assisted by MT corpora. We omitted the baseline model NCLS+MS (Zhu et al., 2019), which uses another separate decoder conducting MS task.

All the models in our experiments can handle both Machine Translation task and Cross-lingual Summarization task. However, models sharing partial parameters possess more parameters than those which shares all parameters between both tasks. Details can be seen in Table 1. Hence, to achieve a fairer comparison, we set up two variations of our model.

- **CSC\textsubscript{base}**: the base model of which the hidden dimension is 512. It has far fewer parameters than the above multitask models, leading to a potentially unfair comparison.
- **CSC\textsubscript{768}**: the enhanced model with a hidden dimension of 768. It owns the similar capacity as multitask models.

To verify the effect of our proposed data augmentation, we also set another variation of our model: **CSC\textsubscript{Multitask}**. Using augmented data as part of the training data, this variation treats CLS, MT, and the Augmented CLS as three separate tasks. We also conduct experiments with its three variations: SD, SE, and SA.

For CSC models except the multitask variation, two different inference modes are tested. One is
We use Fairseq toolkit (Ott et al., 2019) to implement all of our models with the dictionary of mBART (Liu et al., 2020), including 200027 tokens. For model architecture, we follow the base Transformer settings proposed by Vaswani et al. (2017). All the Transformer encoder and decoder layers have the same model size, the proposed CSC model with fixed compression rate achieves the best performance without the any target length information. Moreover, the CSC model with oracle γ achieves even better results, showing that incorporating compression rate into the training brings a significant improvement. For models with same number of parameters, CSC model with fixed compression rate outperforms all the baseline models.

As for the CSC\textsubscript{multitask} models with augmented data, we can see that the performances of these models achieves an improvement over their vanilla versions. Meanwhile, the model CSC\textsubscript{multitask}(SA) also outperforms the CSC model with δ = 0.2 and fixed compression rate. We speculate that the reason is the model is trained with a larger compression rate mode, where a fixed γ is computed by the average γ of CLS training set and fed into the model. In Zh2EnSum Dataset, the average γ is around 0.25. The other is to test the model with the true (or desired) γ of each sample, named as oracle. It can verify the performance of our model when we have controllable signals during inference.

We show that our model outperforms all of the baseline models in all the comparable conditions. Details can be seen in Section 6.

### 5.3 Implementation Details

We use Fairseq toolkit (Ott et al., 2019) to implement all of our models with the dictionary of mBART (Liu et al., 2020), including 200027 tokens. For model architecture, we follow the base Transformer settings proposed by Vaswani et al. (2017). All the Transformer encoder and decoder layers contain 6 layers. The size of feed forward layers are 2048. The hidden size for attention module is either 512 or 768 depending on the model variation. Each attention layer contains 8 different attention heads. The parameter statistics are shown in Table 1.

The Adam optimizer is used to train all the models with learning rate of 5e-4 and 5000 warm-up steps 5000. We use dropout of 0.2 for feed forward layers and 0.1 for attention layers. All the models are trained with 8 Tesla V100 32G GPUs. The batch size is 2048 tokens and the model parameters are updated every 16 batches. We apply the early stop strategy when the validation loss no longer improves for three epochs and choose the checkpoint with the best validation loss to conduct evaluation. During inference, we use beam search with a beam size of 5. Meanwhile, 3-gram blocking is applied to avoid repetition problem. All the hyper-parameters are tuned using the perplexity and validation loss metric on the validation set. For all CSC models, the data among all the different tasks is fed into the model uniformly.

### 6 Experiments

#### 6.1 Automatic Evaluation

We use F1 score of the standard ROUGE metric to evaluate all the models automatically. Results are shown in Table 2.

For each comparable scenario where all the models have the same model size, the proposed CSC model with fixed compression rate achieves the best performance without the any target length information. Moreover, the CSC model with oracle γ achieves even better results, showing that incorporating compression rate into the training brings a significant improvement. For models with same number of parameters, CSC model with fixed compression rate outperforms all the baseline models.

As for the CSC\textsubscript{multitask} models with augmented data, we can see that the performances of these models achieves an improvement over their vanilla versions. Meanwhile, the model CSC\textsubscript{multitask}(SA) also outperforms the CSC model with δ = 0.2 and fixed compression rate. We speculate that the reason is the model is trained with a larger com-

### Table 1: The parameter numbers of different models

| Model                  | Dimension | Parameters |
|------------------------|-----------|------------|
| NCLS+MT (SD)           | 512       | 450M       |
| NCLS+MT (SE)           | 512       | 450M       |
| CSC\textsubscript{multitask} (SD) | 512 | 600M       |
| CSC\textsubscript{multitask} (SE) | 512 | 600M       |
| NCLS+MT (SA)           | 512       | 300M       |
| CSC                    | 512       | 300M       |
|                       | 768       | 460M       |

### Table 2: F1 scores of ROUGE and BERTScore in Zh2EnSum dataset.

| Models                  | Parameters | R-1  | R-2  | R-L  |
|-------------------------|------------|------|------|------|
| NCLS                   |            | 35.60| 16.78| 30.27|
| mBART(Finetune)         |            | 37.14| 18.35| 32.41|
| TETTran                |            | 23.09| 7.33 | 18.74|
| TLTran                 |            | 33.92| 15.81| 29.86|
| TLTran (fixed, δ = 0.2) | ≈300M      | 38.81| 19.96| 33.90|
| TLTran (fixed, δ = 0.2) | ≈450M      | 38.25| 19.60| 33.58|
| CSC\textsubscript{multitask} (SD) | ≈600M | 38.39| 19.85| 33.68|
| CSC\textsubscript{multitask} (SD) | ≈450M | 39.22| 20.10| 34.29|
| CSC\textsubscript{multitask} (SA) | ≈300M | 39.12| 20.18| 34.37|
| CSC\textsubscript{multitask} (SA) | ≈450M | 40.30| 21.43| 35.46|

### Table 3: Results of CSC models trained with different compression rates.

| Models                  | Fixed γ | Oracle γ | Length Variance |
|-------------------------|---------|----------|-----------------|
| R-1 | R-2 | R-L | R-1 | R-2 | R-L |               |
| δ = 0.020  | 36.80 | 18.61 | 32.47 | 40.09 | 20.93 | 35.14 | 0.72  |
| δ = 0.033  | 37.43 | 18.85 | 33.02 | 40.72 | 21.05 | 35.70 | 2.09  |
| δ = 0.050  | 37.70 | 19.27 | 33.06 | 40.59 | 21.23 | 35.71 | 2.74  |
| δ = 0.200  | 38.96 | 20.29 | 34.11 | 40.30 | 21.43 | 35.46 | 20.32 |

R-1, R-2, and R-L represents ROUGE-1, ROUGE-2, and ROUGE-L, respectively.
expression bin, which means that the model, to some extent, loses its controllability during the inference. We will further discuss this phenomenon in Section 6.3.

For the comparison among baselines, we found that models sharing decoder performs better than those models sharing encoder. This has not been well addressed in previous studies as far as we know, which provides a practical tendency for us when using multitask models for the CLS task.

### 6.2 Analysis on Controllability of Compression Rates

Apart from improving the performance of the model, a prominent advantage of the proposed CSC is that, it is able to generate summaries with different target length by adjusting the compression rate. To explore this feature, we generate summaries with different compression rate signals for each document in test set. We show the recall and precision of ROUGE metric in Figure 4. Obviously, as the compression rate raises, the ROUGE recall values also raises, which means that the model is absorbing more information into the final summary. However, since the expected information of the target summary is unchanged, the precision metric drops when the generated summary becomes longer.

We also show the BLEU score with different compression rates where we apply the same model directly on the WMT17 Zh2En test set\(^1\). It clearly shows the transition from CLS task to MT task: as the compression rate goes up, the BLEU score also raises until it completely meets the MT task.

Besides, we measure the average summary lengths of summaries guided by different compression rates. Results are shown in Table 4. We can clearly see that the summary becomes longer while the compression rate goes up.

### 6.3 Analysis on Compression Rate Bin

The bin width \(\delta\) of the compression rate an important hyper-parameter to control the precision. With a smaller \(\delta\), the model can more precisely capture the relationship between the input compression rate and the length ratio of the document-summary pair. Hence, it is intuitive that a smaller \(\delta\) should lead to a better oracle performance and a worse performance given a fixed compression rate, and vice versa.

To verify this hypothesis, we train various models with different bin widths. Results are shown in the Table 3. We can observe that as \(\delta\) become larger, the performance of the model gets better with fixed compression rate. However, the performance of the oracle case does not drop as we expected, so we further measure the variance between the lengths of references and predictions. Results are shown in the rightmost column in Table 3. As \(\delta\) becomes smaller, the variance also becomes smaller, which means the compression rate modeling does predict more accurate length. We speculate that the slightly bad oracle performance is because the model extracts some wrong information.

Above observations motivate us to utilize the CSC training to meet different practical requirements. For example, in the application of news push notifications on different devices (e.g., iPhone and iPad), we need to strictly control the length of the generated summary, then we can use the model trained with small bins. If we only care about the overall quality of the summary rather than an expected length, we can set a relatively large bin.

---

1 We use sacreBLEU implementation (Post, 2018) for BLEU evaluation.

---

| \(\gamma\) | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 | 1.0 |
|------------|-----|-----|-----|-----|-----|-----|
| Avg. Length | 7.45 | 12.90 | 21.65 | 30.95 | 40.63 | 53.14 |

Table 4: The average length of summaries generated with different compression rate.
6.4 Case Study

In this section, we show two case studies. One is emphasizing on the quality, where we can clearly see the summaries generated by the CSC model is better than other models. The other is focusing on the controllability, where results of different compression rates are shown.

6.4.1 Case study on quality

For summary quality, we show system summaries generated by different models in Figure 5. The output of baseline models lost some important information such as ‘the appearance of the car model’ and ‘Chengdu Auto Show’. Some baselines such as NCLS+MT(SD) suffer from fluency issue. Specifically, non-factual outputs are generated by some baselines such ‘2014’, indicating the existence of hallucination problem. In this case, the oracle $\gamma$ and the fixed $\gamma$ are in the same compression bin. Hence, we omit the oracle result. Overall, the summary produced by CSC is the most similar to the reference summary.

6.4.2 Case study on controllability

Guided by different compression rates, summary samples generated by our CSC model ($\delta = 0.02$) are shown in Figure 6. A clear upward trend of summary lengths is observed when the compression rate become larger. When given the compression rate of 0.1, the summary only contains four simple words ‘Husband locks his wife’ which is fairly important for the article. As the compression rate becomes larger, the summary gets longer. Eventually, the generated text becomes the translation of the source text. This case shows the practical controllability of the proposed CSC model. However, the model still has its weakness. For example, when $\gamma = 1.0$, a translation mistake is made by the model. The word ‘车’ is translated into ‘bus’ instead of ‘car’. This shows that the summarization training may have a negative effect to the MT task.

7 Discussion

7.1 Application of CSC

We claim that our proposed CSC task brings a wider image for the application of cross-lingual summarization. With the development of technology, summary requirements for different lengths become more mightiness. For example, one news document may need different versions of summaries: watch/phone notifications, twitter short texts, search result summaries, web page summaries, etc. Our method provides a simple yet effective method to achieve the diversity. With one CSC model, summaries with various lengths can be generated at once (through parallel computing).
fulfilling different requirements. In general, our method is practical in various real scenarios.

7.2 Incorporating the MS task into CSC

Besides the MT task, another high-resource task related to CLS is the Monolingual Summarization task. We can readily incorporate the MS task into CSC framework. In our preliminary experiments we found that generating summaries in its own language may distract the performance of the cross-lingual summary, leading to a worse performance than our proposed approach. However, we think that incorporating the MS task should benefit the CLS task and need more exploration. We will leave it for the future work.

8 Conclusion

In this paper, we propose a novel task, Cross-lingual Summarization with Compression Rate. We propose a simple yet effective data-augment method and a modified Transformer model to bridge the MT and CLS tasks. Experiments demonstrate that our approach not only improve the performance of the CLS task by leveraging MT corpora, but also provides a promising direction for the controllability of summaries in desired lengths. We believe it will have great potentials in real application scenarios.

References

Yu Bai, Yang Gao, and Heyan Huang. 2021. Cross-lingual abstractive summarization with limited parallel resources. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6910–6924, Online. Association for Computational Linguistics.

Yue Cao, Hui Liu, and Xiaojun Wan. 2020. Jointly learning to align and summarize for neural cross-lingual summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6220–6231.

Udo Hahn and Inderjeet Mani. 2000. The challenges of automatic summarization. Computer, 33(11):29–36.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1967–1972.

Faisal Ladghak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. Wikilingua: A new benchmark dataset for multilingual abstractive summarization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 4034–4048.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726–742.

Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4110–4119.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Łukasz Kaiser. 2015. Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06174.

Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global optimization under length constraint for neural text summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1039–1048.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of NAACL-HLT 2019: Demonstrations.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3999–4004.

Sho Takase and Naoaki Okazaki. 2020. Multi-task learning for cross-lingual abstractive summarization. arXiv preprint arXiv:2010.07503.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.
Xiaojun Wan. 2011. Using bilingual information for cross-language document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1546–1555.

Xiaojun Wan, Huiying Li, and Jianguo Xiao. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926.

Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng. 2005. Text summarization using a trainable summarizer and latent semantic analysis. *Information processing & management*, 41(1):75–95.

Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2016. Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1842–1853.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. Ncls: Neural cross-lingual summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3045–3055.

Junnan Zhu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Attend, translate and summarize: An efficient method for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1309–1321.