Abstractive Summarization with Combination of Pre-trained Sequence-to-Sequence and Saliency Models

Abstract

Pre-trained sequence-to-sequence (seq-to-seq) models have significantly improved the accuracy of several language generation tasks, including abstractive summarization. Although the fluency of abstractive summarization has been greatly improved by fine-tuning these models, it is not clear whether they can also identify the important parts of the source text to be included in the summary. In this study, we investigated the effectiveness of combining saliency models that identify the important parts of the source text with the pre-trained seq-to-seq models through extensive experiments. We also proposed a new combination model consisting of a saliency model that extracts a token sequence from a source text and a seq-to-seq model that takes the sequence as an additional input text. Experimental results showed that most of the combination models outperformed a simple fine-tuned seq-to-seq model on both the CNN/DM and XSum datasets even if the seq-to-seq model is pre-trained on large-scale corpora. Moreover, for the CNN/DM dataset, the proposed combination model exceeded the previous best-performed model by 1.33 points on ROUGE-L.

Introduction

Pre-trained language models such as BERT BIBREF0 have significantly improved the accuracy of various language processing tasks. However, we cannot apply BERT to language generation tasks as is because its model structure is not suitable for language generation. Several pre-trained seq-to-seq models for language generation BIBREF1, BIBREF2 based on an encoder-decoder Transformer model, which is a standard model for language generation, have recently been proposed. These models have achieved blackstate-of-the-art results in various language generation tasks, including abstractive summarization.

However, when generating a summary, it is essential to correctly predict which part of the source text should be included in the summary. Some previous studies without pre-training have examined combining extractive summarization with abstractive summarization BIBREF3, BIBREF4. Although pre-trained seq-to-seq models have achieved higher accuracy compared to previous models, it is not clear whether modeling “Which part of the source text is important?” can be learned through pre-training.

blackThe purpose of this study is to clarify the blackeffectiveness of combining saliency models that identify the important part of the source text with a pre-trained seq-to-seq model in the abstractive summarization task. Our main contributions are as follows:

We investigated nine combinations of pre-trained seq-to-seq and token-level saliency models, where the saliency models share the parameters with the encoder of the seq-to-seq model or extract important tokens independently of the encoder.

We proposed a new combination model, the conditional summarization model with important tokens (CIT), in which a token sequence extracted by a saliency model is explicitly given to a seq-to-seq model as an additional input text.

We evaluated the combination models on the CNN/DM BIBREF5 and XSum BIBREF6 datasets. Our CIT model outperformed a simple fine-tuned model in terms of ROUGE scores on both datasets.

Task Definition

Our study focuses on two tasks: abstractive summarization and blacksaliency detection. The main task is abstractive summarization and the sub task is blacksaliency detection, which is the prediction of important parts of the source text. The problem formulations of each task are described below.

Task 1 (Abstractive summarization) Given the source text X, the output is an abstractive summary Y = $(y_1,\ldots ,y_T)$.

Task 2 (Saliency detection) Given the source text X with L words X= $(x_1,\dots ,x_L)$, the output is the saliency score $S = \lbrace S_1, S_2, ... S_L \rbrace $.

In this study, we investigate several combinations of models for these two tasks.

Pre-trained seq-to-seq Model

There are several pre-trained seq-to-seq models applied for abstractive summarization BIBREF7, BIBREF8, BIBREF2. The models use a simple Transformer-based encoder-decoder model BIBREF9 in which the encoder-decoder model is pre-trained on large unlabeled data.

Pre-trained seq-to-seq Model ::: Transformer-based Encoder-Decoder

In this work, we define the Transformer-based encoder-decoder model as follows.

Pre-trained seq-to-seq Model ::: Transformer-based Encoder-Decoder ::: Encoder

The encoder consists of M layer encoder blocks. The input of the encoder is $X = \lbrace x_i, x_2, ... x_L \rbrace $. The output through the M layer encoder blocks is defined as

The encoder block consists of a self-attention module and a two-layer feed-forward network.

Pre-trained seq-to-seq Model ::: Transformer-based Encoder-Decoder ::: Decoder

The decoder consists of M layer decoder blocks. The inputs of the decoder are the output of the encoder H_e^M and the output of the previous step of the decoder $\lbrace y_1,...,y_{t-1} \rbrace $. The output through the M layer Transformer decoder blocks is defined as

In each step t, the h_{dt}^M is projected to blackthe vocabulary space and the decoder outputs the highest probability token as the next token. The Transformer decoder block consists of a self-attention module, a context-attention module, and a two-layer feed-forward network.

Pre-trained seq-to-seq Model ::: Transformer-based Encoder-Decoder ::: Multi-head Attention

The encoder and decoder blocks use multi-head attention, which consists of a combination of K attention heads and is denoted as $\mathrm {Multihead}(Q, K, V) = \mathrm {Concat}(\mathrm {head}_1, ...,\mathrm {head}_k)W^o$, where each head is $\mathrm {head}_i = \mathrm {Attention}(QW_i^Q, KW_i^K , VW_i^V)$.

The weight matrix A in each attention-head $\mathrm {Attention}(\tilde{Q}, \tilde{K}, \tilde{V}) = A \tilde{V}$ is defined as

where $d_k = d / k$, $\tilde{Q} \in \mathbb {R}^{I \times d}$, $\tilde{K}, \tilde{V} \in \mathbb {R}^{J \times d}$.

In the m-th layer of self-attention, the same representation $H^m_{\cdot }$ is given to Q, K, and V. In the context-attention, we give H^m_d to Q and H^M_e to K and V.

Pre-trained seq-to-seq Model ::: Summary Loss Function

To fine-tune the seq-to-seq model for abstractive summarization, we use cross entropy loss as

where N is the number of training samples.

Saliency Models

Several studies have proposed the combination of a token-level saliency model and a seq-to-seq model, blackwhich is not pre-trained, and reported its effectiveness BIBREF3, BIBREF10. We also use a simple token-level saliency model blackas a basic model in this study.

Saliency Models ::: Basic Saliency Model

A basic saliency model consists of M-layer Transformer encoder blocks ($\mathrm {Encoder}_\mathrm {sal}$) and a single-layer feed-forward network. We define the saliency score of the l-th token ($1 \le l \le L$) in the source text as

where ${\rm Encoder_{sal}()}$ represents the output of the last layer of black$\rm Encoder_{sal}$, $W_1 \in \mathbb {R}^{d}$ and b_1 are learnable parameters, and $\sigma $ represents a sigmoid function.

Saliency Models ::: Two Types of Saliency Model for Combination

In this study, we use two types of saliency model for combination: a shared encoder and an extractor. Each model structure is based on the basic saliency model. We describe them below.

Saliency Models ::: Two Types of Saliency Model for Combination ::: Shared encoder

The shared encoder blackshares the parameters of black$\rm Encoder_{sal}$ and the encoder of the seq-to-seq model. This model is jointly trained with the seq-to-seq model and the saliency score is used to bias the representations of the seq-to-seq model.

Saliency Models ::: Two Types of Saliency Model for Combination ::: Extractor

The extractor extracts the important tokens or sentences from the source text on the basis of the saliency score. The extractor is separated with the seq-to-seq model, and each model is trained independently.

Saliency Models ::: Pseudo Reference Label

The saliency model predicts blackthe saliency score S_l for each token x_l. If there is a reference label r_l $\in \lbrace 0, 1\rbrace $ for each x_l, we can train the saliency model in a supervised manner. However, the reference label for each token is typically not given, since the training data for the summarization consists of only the source text and its reference summary. Although there are no reference saliency labels, we can make pseudo reference labels by aligning both source and summary token sequences and extracting common tokens BIBREF3. blackWe used pseudo labels when we train the saliency model in a supervised manner.

Saliency Models ::: Saliency Loss Function

To train the saliency model in a supervised way blackwith pseudo reference labels, we use binary cross entropy loss as

where r_l^n is a pseudo reference label of token x_l in the n-th sample.

Combined Models

This section describes nine combinations of the pre-trained seq-to-seq model and saliency models.

Combined Models ::: Combination types

We roughly categorize the combinations into three types. Figure FIGREF23 shows an image of each combination.

The first type uses the shared encoder (§SECREF26). These models consist of the shared encoder and the decoder, where the shared encoder module blackplays two roles: saliency detection and the encoding of the seq-to-seq model. blackThe saliency scores are used to bias the representation of the seq-to-seq model for several models in this type.

The second type uses the extractor (§SECREF34, §SECREF37). These models consist of the extractor, encoder, and decoder and follow two steps: first, blackthe extractor blackextracts the important tokens or sentences from the source text, and second, blackthe encoder uses them as an input of the seq-to-seq models. Our proposed model (CIT) belongs to this type.

The third type uses both the shared encoder and the extractor (§SECREF39). These models consist of the extractor, shared encoder, and decoder and also follow two steps: first, blackthe extractor extracts the important tokens from the source text, and second, blackthe shared encoder uses them as an input of the seq-to-seq model.

Combined Models ::: Loss function

From the viewpoint of the loss function, there are two major types of model: those that use the saliency loss (§SECREF21) and those that do not. We also denote the loss function for the seq-to-seq model as $L_\mathrm {abs}$ and the loss function for the extractor as $L_\mathrm {ext}$. black$L_\mathrm {ext}$ is trained with $L_\mathrm {sal}$, and $L_\mathrm {abs}$ is trained with $L_\mathrm {sum}$ or $L_\mathrm {sum} + L_\mathrm {sal}$.

Combined Models ::: Using Shared Encoder to Combine the Saliency Model and the Seq-to-seq Model ::: Multi-Task (MT)

This model trains the shared encoder and the decoder by minimizing both the summary and saliency losses. The loss function of this model is $L_\mathrm {abs} = L_\mathrm {sum} + L_\mathrm {sal}$.

Combined Models ::: Using Shared Encoder to Combine the Saliency Model and the Seq-to-seq Model ::: Selective Encoding (SE)

This model uses the saliency score to weight the shared encoder output. Specifically, the final output h_{el}^M of the shared encoder is weighted as

Then, we replace the input of the decoder h_{el}^M with $\tilde{h}_{el}^{M}$. Although BIBREF10 used BiGRU, we use Transformer for fair comparison. The loss function of this model is $L_\mathrm {abs} = L_\mathrm {sum}.$

Combined Models ::: Using Shared Encoder to Combine the Saliency Model and the Seq-to-seq Model ::: Combination of SE and MT

This model has the same structure as the SE. The loss function of this model is $L_\mathrm {abs} =

L_\mathrm {sum} + L_\mathrm {sal}$.

Combined Models ::: Using Shared Encoder to Combine the Saliency Model and the Seq-to-seq Model ::: Selective Attention (SA)

This model weights the attention scores of the decoder side, unlike the SE model. Specifically, the attention score $a_{i}^{t} \in \mathbb {R}^L$ in each step t is weighted by S_l. a_{i}^{t} is a t-th row of $A_i \in \mathbb {R}^{T \times L}$, which is a weight matrix of the i-th attention head in the context-attention (Eq. (DISPLAY_FORM12)).

BIBREF3 took a similar approach in that their model weights the copy probability of a pointer-generator model. However, as the pre-trained seq-to-seq model does not have a copy mechanism, we weight the context-attention for all Transformer decoder blocks. The loss function of this model is $L_\mathrm {abs} = L_\mathrm {sum}$.

Combined Models ::: Using Shared Encoder to Combine the Saliency Model and the Seq-to-seq Model ::: Combination of SA and MT

This model has the same structure as the SA. The loss function of this model is $L_\mathrm {abs} = L_\mathrm {sum} + L_\mathrm {sal}$.

Combined Models ::: Using the Extractor to Refine the Input Text ::: Sentence Extraction then Generation (SEG)

This model first extracts the saliency sentences on the basis of a sentence-level saliency score S_j. S_j is calculated by using the token level saliency score of the extractor, S_l, as where N_j and X_j are the number of tokens and the set of tokens within the j-th sentence. Top P sentences are extracted according to the sentence-level saliency score and then concatenated as one text X_s. These extracted sentences are then used as the input of the seq-to-seq model.

In the training, we extracted X_s, which maximizes the ROUGE-L scores with the reference summary text. In the test, we used the average number of sentences in X_{s} in the training set as P. The loss function of the extractor is $L_\mathrm {ext} = L_\mathrm {sal}$, and that of the seq-to-seq model is $L_\mathrm {abs} = L_\mathrm {sum}$.

Combined Models ::: Proposed: Using Extractor to Extract an Additional Input Text ::: Conditional Summarization Model with Important Tokens

We propose a new combination of the extractor and the seq-to-seq model, CIT, which can consider important tokens explicitly. Although the SE and SA models softly weight the representations of the source text or attention scores, they cannot select salient tokens explicitly. SEG explicitly extracts the salient sentences from the source text, but it cannot give token-level information to the seq-to-seq model, and it sometimes drops important information when extracting sentences. In contrast, CIT uses the tokens extracted according to saliency scores as an additional input of the seq-to-seq model. By adding token-level information, CIT can effectively guide the abstractive summary without dropping any important information.

Specifically, K tokens $C = \lbrace c_1, ..., c_K \rbrace $ are extracted in descending order of the saliency score S. S is obtained by inputting X to the extractor. The order of C retains the order of the source text X. A combined text $\tilde{X} = \mathrm {Concat}(C, X)$ is given to the seq-to-seq model as the input text. The loss function of the extractor is $L_\mathrm {ext} = L_\mathrm {sal}$, and that of the seq-to-seq model is $L_\mathrm {abs} = L_\mathrm {sum}$.

Combined Models ::: Proposed: Combination of Extractor and Shared Encoder ::: Combination of CIT and SE

This model combines the CIT and SE, so CIT uses an extractor for extracting important tokens, and SE is trained by using a shared encoder in the seq-to-seq model. The SE model is trained in an unsupervised way. The output $H^M_e \in \mathbb {R}^{L+K}$ of the shared encoder is weighted by saliency score $S \in \mathbb {R}^{L+K}$ with Eq. (DISPLAY_FORM29), where S is estimated by using the output of the shared encoder with Eq. (DISPLAY_FORM16). The loss function of the extractor is $L_\mathrm {ext} = L_\mathrm {sal}$, and that of the seq-to-seq model is $L_\mathrm {abs} = L_\mathrm {sum}$.

Combined Models ::: Proposed: Combination of Extractor and Shared Encoder ::: Combination of CIT and SA

This model combines the CIT and SA, so we also train two saliency models. The SA model is trained in an unsupervised way, the same as the CIT + SE model. The attention score $a_i^t \in \mathbb {R}^{L+K}$ is weighted by $S \in \mathbb {R}^{L+K}$ with Eq. (DISPLAY_FORM32). The loss function of the extractor is $L_\mathrm {ext} = L_\mathrm {sal}$, and that of the seq-to-seq model is $L_\mathrm {abs} = L_\mathrm {sum}$.

Experiments ::: Dataset

We used the CNN/DM dataset BIBREF5 and the XSum dataset BIBREF6, which are both standard datasets for news summarization. The details of the two datasets are listed in Table TABREF48. The CNN/DM is a highly extractive summarization dataset and the XSum is a highly abstractive summarization dataset.

Experiments ::: Dataset ::: Model Configurations

blackWe used BART$_{\mathrm {LARGE}}$ BIBREF1, which is one of the state-of-the-art models, as the pre-trained seq-to-seq model and RoBERTa$_\mathrm {BASE}$ BIBREF11 as the initial model of the extractor. In the extractor of CIT, stop words and duplicate tokens are ignored for the XSum dataset.

We used fairseq for the implementation of the seq-to-seq model. For fine-tuning of BART$_\mathrm {LARGE}$ and the combination models, we used the same parameters as the official code. For fine-tuning of RoBERTa$_\mathrm {BASE}$, we used Transformers. We set the learning rate to 0.00005 and the batch size to 32.

Experiments ::: Evaluation Metrics

We used ROUGE scores (F1), including ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L), as the evaluation metrics BIBREF12. ROUGE scores were calculated using the files2rouge toolkit.

Experiments ::: Results ::: Do saliency models improve summarization accuracy in highly extractive datasets?

Rouge scores of the combined models on the CNN/DM dataset are shown in Table TABREF51. We can see that all combined models outperformed the simple fine-tuned BART. This indicates that the saliency detection is effective in highly extractive datasets. One of the proposed models, CIT + SE, achieved the highest accuracy. The CIT model alone also outperformed other saliency models. This indicates that the CIT model effectively guides the abstractive summarization by combining explicitly extracted tokens.

Experiments ::: Results ::: Do saliency models improve summarization accuracy in highly abstractive

datasets?

Rouge scores of the combined models on the XSum dataset are shown in Table TABREF52. The CIT model performed the best, although its improvement was smaller than on the CNN/DM dataset. Moreover, the accuracy of the MT, SE + MT, and SEG models decreased on the XSum dataset. These results were very different from those on the CNN/DM dataset.

One reason for the difference can be traced to the quality of the pseudo saliency labels. CNN/DM is a highly extractive dataset, so it is relatively easy to create token alignments for generating pseudo saliency labels, while in contrast, a summary in XSum is highly abstractive and short, which makes it difficult to create pseudo labels with high quality by simple token alignment. To improve the accuracy of summarization in this dataset, we have to improve the quality of the pseudo saliency labels and the accuracy of the saliency model.

Experiments ::: Results ::: How accurate are the outputs of the extractors?

We analyzed the quality of the tokens extracted by the extractor in CIT. The results are summarized in Table TABREF55. On the CNN/DM dataset, the ROUGE-1 and ROUGE-2 scores of our extractor (Top-K tokens) were higher than other models, while the ROUGE-L score was lower than the other sentence-based extraction method. This is because that our token-level extractor finds the important tokens whereas the seq-to-seq model learns how to generate a fluent summary incorporating these important tokens.

On the other hand, the extractive result on the XSum dataset was lower. For highly abstractive datasets, there is little overlap between the tokens. We need to consider how to make the high-quality pseudo saliency labels and how to evaluate the similarity of these two sequences.

Experiments ::: Results ::: Does the CIT model outperform other fine-tuned models?

Our study focuses on the combinations of saliency models and the pre-trained seq-to-seq model. However, there are several studies that focus more on the pre-training strategy. We compared the CIT model with those models. Their ROUGE scores are shown in Tables TABREF57 and TABREF58. From Table TABREF57, we can see that our model outperformed the recent pre-trained models on the CNN/DM dataset. Even though PEGASUS$_\mathrm {HugeNews}$ was pre-trained on the largest corpus comprised of news-like articles, the accuracy of abstractive summarization was not improved much. Our model improved the accuracy without any additional pre-training. This result indicates that it is more effective to combine saliency models with the seq-to-seq model for generating a highly extractive summary.

On the other hand, on the XSum dataset, PEGASUS$_\mathrm {HugeNews}$ improved the ROUGE scores and achieved the best results. In the XSum dataset, summaries often include the expressions that are not written in the source text. Therefore, increasing the pre-training data and learning more patterns were effective. However, by improving the quality of the pseudo saliency labels, we should be able to improve the accuracy of the CIT model.

Related Work and Discussion ::: Pre-trained Language Models for Abstractive Summarization

BIBREF18 used BERT for their sentence-level extractive summarization model. BIBREF19 proposed a new pre-trained model that considers document-level information for sentence-level extractive summarization. Several researchers have published pre-trained encoder-decoder models very recently BIBREF20, BIBREF1, BIBREF2. BIBREF20 pre-trained a Transformer-based pointer-generator model. BIBREF1 pre-trained a standard Transformer-based encoder-decoder model using large unlabeled data and achieved state-of-the-art results. BIBREF8 and BIBREF16 extended the BERT structure to handle seq-to-seq tasks.

All the studies above focused on how to learn a universal pre-trained model; they did not consider the combination of pre-trained and saliency models for an abstractive summarization model.

Related Work and Discussion ::: Abstractive Summarization with Saliency Models

BIBREF4, BIBREF3, and BIBREF21 incorporated a sentence- and word-level extractive model in the pointer-generator model. Their models weight the copy probability for the source text by using an extractive model and guide the pointer-generator model to copy important words. BIBREF22 proposed a keyword guided abstractive summarization model. BIBREF23 proposed a sentence extraction and re-writing model that is trained in an end-to-end manner by using reinforcement learning. BIBREF24 proposed a search and rewrite model. BIBREF25 proposed a combination of sentence-level extraction and compression. None of these models are based on a pre-trained model. In contrast, our purpose is to clarify whether combined models are effective or not, and we are the first to investigate the combination of pre-trained seq-to-seq and saliency models. We compared a variety of combinations and clarified which combination is the most effective.

Conclusion

This is the first study that has conducted extensive experiments to investigate the effectiveness of incorporating saliency models into the pre-trained seq-to-seq model. From the results, we found that saliency models were effective in finding important parts of the source text, even if the seq-to-seq model is pre-trained on large-scale corpora, especially for generating an highly extractive summary. We also proposed a new combination model, CIT, that outperformed simple fine-tuning and other combination models. Our combination model improved the summarization accuracy without any additional pre-training data and can be applied to any pre-trained model. While recent studies have been conducted to improve summarization accuracy by increasing the amount of pre-training data and developing new pre-training strategies, this study sheds light on the importance of saliency models in abstractive summarization.