Abstractive Document Summarization with Word Embedding Reconstruction

Jingyi You[1], **Chenlong Hu**[1], **Hidetaka Kamigaito**[1],
Hiroya Takamura[2] **and Manabu Okumura**[1]
[1]Tokyo Institute of Technology
[2]National Institute of Advanced Industrial Science and Technology (AIST)
{youjy,huchenlong,kamigaito,oku}@lr.pi.titech.ac.jp
takamura.hiroya@aist.go.jp

Abstract

Neural sequence-to-sequence (Seq2Seq) models and BERT have achieved substantial improvements in abstractive document summarization without and with pre-training, respectively. However, they sometimes repeatedly attend to unimportant source phrases while mistakenly ignore important ones. We present new *reconstruction* mechanisms on two levels to alleviate this issue. The sequence-level reconstructor reconstructs the whole source document from the hidden layer of the target summary, while the word embedding-level one rebuilds the average of word embeddings of the source at the target side to guarantee that as much critical information is included in the summary as possible. Based on the assumption that inverse document frequency (IDF) measures how important a word is, we further leverage the IDF weights in our embedding-level reconstructor. The proposed frameworks lead to promising improvements for ROUGE metrics and human rating on both the CNN/Daily Mail and Newsroom summarization datasets.

1 Introduction

Single document summarization is designed to automatically compress a document into its short version without changing the main idea. The summarization task is generally divided into two categories: *extractive* methods that copy certain sentences or phrases directly from the source text, and *abstractive* methods that paraphrase the source text by using novel words. Abstractive summarization has the potential to produce summaries in the same way that humans do.

Recent years have witnessed significant progress in the abstractive summarization task performed by Seq2Seq models, which encode a source text and decode its summary. A hybrid of the extractive and abstractive techniques, called pointer-generator network (PGN) (Gu et al., 2016; See et al., 2017), has been widely used as a basis for many studies (Gehrmann et al., 2018; Shen et al., 2019; Shi et al., 2019) thanks to its capability of copying words from the source document and generating new words. With the rich contextual representations, pre-trained encoders (Devlin et al., 2018) have also improved the state-of-the-art on this task (Liu and Lapata, 2019; Raffel et al., 2020). However, the conventional PGN and BERT face two main problems: 1) When generating summaries, PGN tends to frequently pay attention to some source parts while neglecting other parts, which are regarded as over- and under-attention respectively. Therefore, the model requires a mechanism to make sure that as much salient information is transformed from the source to the target as possible. 2) Additionally, BERT is pre-trained for sentences (Xu et al., 2020), thus it is deficient in distinguishing key points from non-key points within long-range dependencies, which leads to inconsequential phrases or words being copied into the summary. This is mainly due to the lack of ability to recognize topic-signifying words from the document (Baziotis et al., 2019).

Similar problems have been encountered in neural machine translation (NMT) (Tu et al., 2016). Tu et al. (2017) developed an encoder-decoder-reconstructor framework with a two-step process that first translates a sentence into another language and then reconstructs the translation back to the source language. The newly added reconstructor rewards the model when it correctly reconstructs the input source sentence from the decoder hidden layer, which forces the salient information to be transferred from the source to the target. Baziotis et al. (2019) applied this concept to unsupervised sentence compression, where the model consists of two encoder-decoder pairs so that no large text-summary dataset is needed.

Inspired by the above, we propose three *recon-*

Proceedings of Recent Advances in Natural Language Processing, pages 1586–1596
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_178

struction methods for neural document summarization. The intuitive approach is a reconstructor that rebuilds the source document from the decoder hidden layer in an encoder-decoder architecture that assigns the corresponding likelihood as a reconstruction loss. When the reconstructed text is significantly different from the original one due to the model incorrectly omitting or repeating some parts to generate the summary, the reconstructor penalizes it during training. We refer to this approach as sequence-level reconstruction and choose it as our first implementation of the idea.

In contrast to the sequence-level reconstruction, reconstructing the word embeddings of source articles would be a rather simple and effective method for document summarization. The embedding-level reconstruction alleviates the length discrepancy between articles and summaries by calculating the distance between the average word embeddings of the source and target sides. We therefore propose the average word embedding reconstructor as our second reconstruction method.

We also present the third reconstructor that focuses on the saliency of words. Generally, certain words that appear frequently in documents have little importance and should not be kept in the summary. In contrast, words that signify the main idea are supposed to be maintained in the summary. As inverse document frequency (IDF) can measure how important a token is and extract salient information (Salton et al., 1983), we reconstruct the IDF-weighted word embeddings of the source at the target to keep topic-signifying words in the summary. Since the summary has far fewer tokens than the document, it is not appropriate to directly incorporate the term frequency (TF) value into our reconstructor. We therefore consider the IDF as the weight of embeddings, rather than the TF-IDF.

In this work, we mainly adopt the PGN as baseline and incorporate the above three reconstructors upon it. We then assign a loss for each reconstructor and leverage each of them as a complement to the baseline objective. On one hand, the reconstruction objectives facilitate the model to generally focus on the entire text rather than parts of it, thereby avoiding under-attention. On the other hand, the IDF-weighted method serves as a selector by giving more weights to topic-signifying words such that it can identify essential parts (Table 1) and prevent over-attention to less important words.

We performed experiments on two datasets and compared our methods with the baselines. Experimental results on the Newsroom dataset demonstrate that we outperformed the baselines by more than 2 points in ROUGE-1, 2, and L. Our methods also led to significant improvements on the CNN/Daily Mail dataset.

Source Document (cnn) a mammoth fire broke out friday morning in a **kentucky** industrial park, sending plumes of thick smoke over the area as authorities worked to contain the damage. the blaze began shortly before 7 a.m. **at the general electric appliance park in louisville**, according to mike weimer from the city 's emergency management agency. he said that there were **no reports of anyone injured or trapped**. video showed both smoke and bright orange flames. **firefighters took up positions around the affected buildings, spraying water from the periphery**. weimer told cnn that authorities didn't know what had caused the fire, which had gone to at least four alarms. according to a ge website, its facility in the louisville appliance park is "revitalizing manufacturing in the united states." the park is large, such that 34 football fields could fit in one of its warehouses in the facility.
Reference fire breaks out **at the general electric appliance park in louisville, kentucky**. city official: **no is believed to be injured or trapped**.
Pointer-Generator the blaze began shortly before 7 a.m. **at the general electric appliance park in louisville**. authorities didn't know what had caused the fire, which had gone to at least four alarms.
Pointer-Generator with IDF-weighted Embedding Reconstruction a mammoth fire broke out friday morning in a **kentucky** industrial park. the blaze began shortly before 7 a.m. **at the general electric appliance park in louisville**. **no reports of anyone injured or trapped**. **firefighters took up positions around the affected buildings**.

Table 1: Example summaries without and with IDF-weighted embedding reconstruction. The original article contains four important pieces of information, expressed in four colors. The summary generated by our method covers all information, while the baseline summary contains only one of them.

2 Preliminaries

In the PGN, a document x that consists of a sequence of tokens $x = \{x_1, x_2, ..., x_I\}$ is fed into a bidirectional LSTM encoder, producing a sequence of hidden states h_i. Then a unidirectional LSTM decoder generates its corresponding summary $y = \{y_1, y_2, ..., y_T\}$ word by word with the limitation of $T \ll I$. I and T indicate the lengths of the source article and the summary, respectively.

The PGN adopts the attention mechanism to learn the alignment and yield target tokens simultaneously. Conditioned on decoder hidden state s_t and context vector c_t for the t-th decoding step, vocabulary distribution P_{vocab} is as follows:

$$P_{vocab,t} = softmax(W_s s_t + W_c c_t + b_{s2s}), \quad (1)$$

$$c_t = \sum_{i=1}^{I} \alpha_{t,i} h_i, \quad (2)$$

$$\alpha_{t,i} = \exp(\alpha'_{t,i}) / \sum_{k=1}^{I} \exp(\alpha'_{t,k}), \quad (3)$$

$$\alpha'_{t,i} = W_\alpha \tanh(W_h h_i + W_{s'} s_t + b_\alpha), \quad (4)$$

where W_s, W_c, W_α, W_h, $W_{s'}$, b_{s2s}, b_α are trainable parameters, and $\alpha_t = \{\alpha_{t,1}, ..., \alpha_{t,I}\}$ is the

attention distribution over the source hidden states.

The PGN additionally employs the copy mechanism to decide whether to copy a word from the source document or to generate a new word through soft switch $p_{gen,t} \in [0, 1]$. $p_{gen,t}$ can be obtained by a feed-forward network whose inputs are the context vector and the encoder and decoder hidden states. The copy distribution is sampled from attention distribution α_t, that is, the copy probability of a source token w is calculated as the sum of attentions towards all occurrences of w. Thus, the joint distribution is calculated as

$$P(y_t) = p_{gen,t} \times P_{vocab,t}(y_t) + (1 - p_{gen,t}) \times \sum_{i:x_i=y_t} \alpha_{t,i}. \quad (5)$$

During training, we use the negative log-likelihood as the loss function:

$$\mathcal{L}_{PGN} = -\sum_{t=1}^{T} \log P(y_t). \quad (6)$$

3 Proposed Model

We next describe the details of our proposed methods, which can be split into two main components:

- *Pointer-generator*: a neural Seq2Seq framework with the attention and copy mechanisms, as introduced in Sec. 2.
- *Reconstructor*: a module that manages to reconstruct the salient information of the original document in its summary. We put forward three independent reconstructors that will be explained in the next subsections. Moreover, our proposed approaches are applicable to any attention-based Seq2Seq summarization architectures.

3.1 Sequence-level Reconstruction

The first reconstructor, as shown in Fig. 1, is expected to recover the full input sequence from the decoded summary, i.e., to reconstruct the one-hot representations of the tokens in the source document to reward the summary with the complete source information. Specifically, the reconstructor generates a reconstructed sequence $\hat{x} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_I\}$ word by word from decoded summary sequence y and decoder hidden state s_t. We obtain a probability distribution over the vocabulary through reconstructor hidden state $\hat{h}_i$ and inverse context vector $\hat{c}_i$:

$$\hat{P}_{vocab,i} = softmax(\hat{W}_h \hat{h}_i + \hat{W}_c \hat{c}_i + \hat{b}_{s2s}), \quad (7)$$

$$\hat{c}_i = \sum_{t=1}^{T} \hat{\alpha}_{i,t} s_t, \quad (8)$$

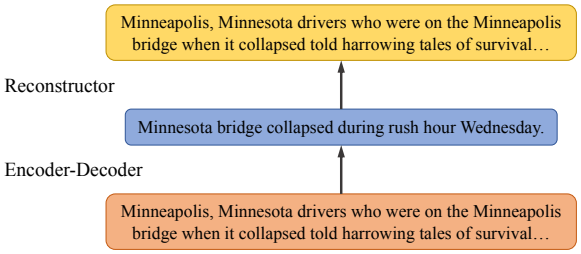

Figure 1: Sequence-level reconstruction model.

where inverse attention $\hat{\alpha}_i$ for reconstructing step i has the same structure as the original attention, except taking the decoder and reconstructor hidden states as inputs and owning independent weighted vectors. Then, we try to minimize the reconstruction loss, which is the negative log-likelihood assigned by $\hat{x}$ to the original document x:

$$\mathcal{L}_{recon} = -\sum_{i=1}^{I} \log \hat{P}_{vocab,i}(\hat{x}_i = x_i). \quad (9)$$

In general, the reconstruction phase can be treated as an inverse process of a standard encoder-decoder.

It is obviously impossible to reduce the loss to a low level because a summary must contain fewer tokens and less information than its original text. However, we can reasonably expect the reconstruction loss to urge the encoder-decoder to embed complete information of the source document.

3.2 Word Embedding-level Reconstruction

In order to ensure the generated summary maintains a similar sequence representation with the source article, we compute the average word embeddings of x and y, and attempt to minimize their cosine distance. For a source word x_i in the input document, we simply utilize vector e_{x_i} obtained from the encoder embedding layer. To represent token y_t, we first concatenate context vector c_{t-1} and embedding $e_{y_{t-1}}$ of the word generated at the previous step. $e_{y_{t-1}}$ keeps and shares part of the word embedding information at the input side, while c_{t-1} simultaneously adds new information about context at the output side. Then, a linear transformation is applied to combine above two vectors:

$$e_{y_t} = W_e \left[c_{t-1}; e_{y_{t-1}}\right] + b_e, \quad (10)$$

where W_e and b_e are trainable parameters.

Considering the difference of the embedding representations between the document and its summary, we take the following actions: 1) the embedding matrix is shared between the encoder and

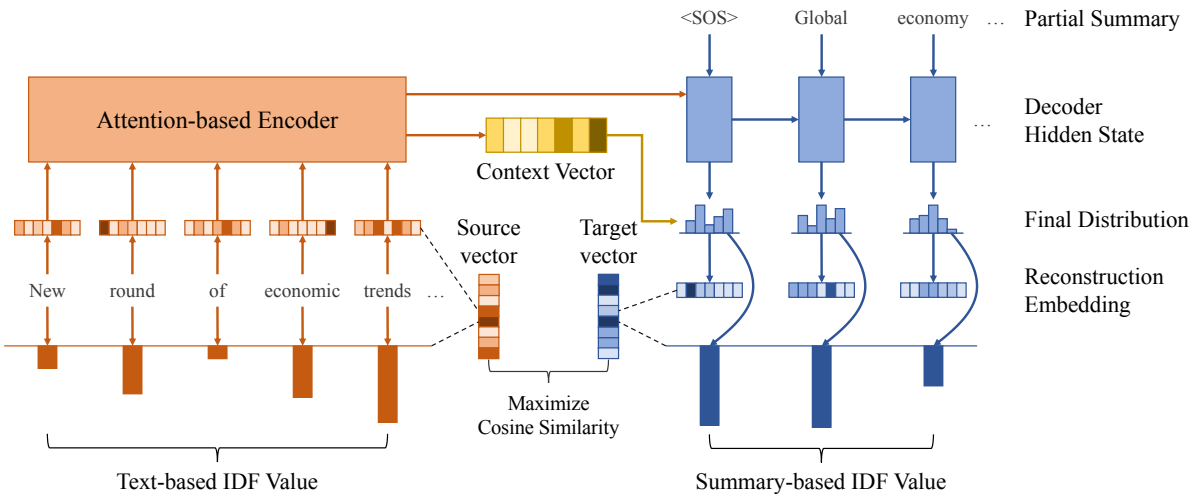

Figure 2: IDF-weighted word embedding reconstruction model.

decoder for unity, 2) the method pays *fair* attention (weight) to each word in the sequence to restrain under-attention, and 3) to prevent the effect of length difference, we calculate the average of the word embeddings for each sequence, i.e., divide the sum by their respective lengths.

We calculate the embedding representations for original article x and summary y as follows:

$$r(x) = \frac{1}{I}\sum_{i=1}^{I} e_{x_i}, r(y) = \frac{1}{T}\sum_{t=1}^{T} e_{y_t}. \quad (11)$$

The reconstruction loss is then examined by the cosine similarity between summary representation $r(y)$ and source representation $r(x)$ accordingly as

$$\mathcal{L}_{recon} = 1 - \cos(r(x), r(y)). \quad (12)$$

3.3 IDF-weighted Embedding-level Reconstruction

The abstractive summarization task is intended to remove duplicate or unimportant words and to paraphrase the rest. The second approach, introduced above, takes the average of word embeddings (i.e., *fair* weights) as the goal of reconstruction. However, attending to all words equally does not suit the objective of this task. Thus, we propose an advanced version of the word embedding reconstruction by incorporating IDFs, as shown in Fig. 2.

Intuitively, some words, e.g., *"the"*, appear in many documents, while others, e.g., *"Harry Potter"*, are not so frequent. Therefore, words with lower IDF values usually have no specific meaning and can be omitted without confusing the main idea. Conversely, higher valued words might signify the topic of an article. This assumption allows the model to distinguish key points from non-key points, thereby avoid over-attention to less important parts. Consequently, IDF-weighted embeddings are used as an alternative, changing Eq. 11 to

$$r(x) = \frac{\sum_{i=1}^{I} IDF_{x_i} e_{x_i}}{\sum_{m=1}^{I} IDF_{x_m}}, r(y) = \frac{\sum_{t=1}^{T} IDF_{y_t} e_{y_t}}{\sum_{n=1}^{T} IDF_{y_n}}. \quad (13)$$

The same form of reconstruction loss as Eq. 12 is implemented here as well.

The IDF values can be computed on the basis of the training dataset. They are calculated separately from a corpus of original articles and reference summaries to create source and target-side dictionaries, respectively. Given a token w, its inverse document frequency can be obtained by

$$IDF_w = \log \frac{1 + n_d}{1 + DF(d, w)} + 1, \quad (14)$$

where n_d denotes the total number of documents in the corpus and $DF(d, w)$ is the number of documents where w appears. 1) At the source side, the model takes the encoder input x_i as a key to search for IDF_{x_i} from the source-side dictionary. 2) Whereas at the target side, the decoder outputs a probability distribution at each time step according to Eq. 5. We choose the word with the highest probability as the output and look for its corresponding IDF_{y_t} in the target-side dictionary.

3.4 Training Loss

We use λ as a hyperparameter to balance $\mathcal{L}_{PGN}$ with $\mathcal{L}_{recon}$. The overall loss can be defined as

$$\mathcal{L} = \mathcal{L}_{PGN} + \lambda \mathcal{L}_{recon}. \quad (15)$$

4 Experiments

4.1 Datasets and Settings

Datasets We carried out the experiments on two benchmark datasets, namely CNN/Daily Mail (Nallapati et al., 2016) and Newsroom (Grusky et al., 2018). Using these two datasets is challenging since we need to compress long news articles into short multi-sentence summaries. To split the CNN/Daily Mail dataset into training/validation/test sets, we followed See et al. (2017) to use the non-anonymized version. We replicated the pre-processing steps released by Shi et al. (2019) to generate the splits of the Newsroom dataset. The basic statistics of the datasets, including the splitting details, are summarized in Table 2. In both datasets, the document and summary were truncated to 400 and 100 tokens, respectively. Additionally, $50K$ of the most frequently occurring tokens in the training dataset were selected to form a vocabulary for both the source and target.

Evaluation metrics We evaluated our models with the ROUGE metrics (Lin, 2004), which compare model-generated summaries with reference summaries by referring to the overlap of unigram (ROUGE-1), bigram (ROUGE-2), and longest common subsequence (ROUGE-L).

Experimental settings We trained our models on a single GeForce RTX 2080Ti GPU (11GB RAM). 128-dimensional word embeddings with random initialization were fine-tuned during training. We utilized a single-layer bidirectional LSTM for the encoder and a unidirectional LSTM for the decoder. Both the encoder and decoder have 256-dimensional hidden states. As for the optimizer, Adam (Kingma and Ba, 2015) with a learning rate of 0.0001 and an initial accumulator value of 0.1 was used. The maximum norm of gradient clipping was set to 2.0. We set the batch size to 8 on the CNN/Daily Mail dataset whereas 32 on the Newsroom dataset. Summaries were decoded through beam search with a beam size of 4 at test time. The maximum iterations on CNN/Daily Mail and Newsroom were $500,000$ and $450,000$, which are both approximately equal to 14 epochs. The same settings were applied to the baselines for comparison.

Training our embedding-level reconstruction model on the CNN/Daily Mail dataset took 41.6 hours, while it took 34.7 hours on the Newsroom dataset. We noticed that embedding-based approaches do not increase training time significantly compared to the baselines. Our approach can improve the performance without introducing too many parameters or sacrificing training efficiency.

When setting the scaling factor λ (in Eq. 15), we found that the baselines with embedding-level reconstructors (in Sec. 3.2, 3.3) achieved the best results when the reconstruction loss was weighted to $\lambda = 2.0$ on the CNN/Daily Mail validation dataset and $\lambda = 2.5$ on the Newsroom validation dataset. However, the sequence-level reconstructor (in Sec. 3.1) worked best when λ was set to 0.1.

Baselines We employed the following excellent baselines for comparison and to demonstrate that our approaches can be transplanted to various Seq2Seq models. As our original intention to design the reconstructors, the **PGN** in Sec. 2 was treated as our main baseline for both datasets. We also adopted **PGN+Coverage** and **PreSumm** (Liu and Lapata, 2019)[1] on the CNN/Daily Mail dataset to further examine the adaptability of our reconstructors. The coverage mechanism (Tu et al., 2016; See et al., 2017) maintains a vector, i.e., a sum of attention distributions over all the former decoding steps, to prevent repeating words. PreSumm employs and fine-tunes the pre-trained context representations of BERT (Devlin et al., 2018) as an encoder. On the Newsroom dataset, we additionally utilized **LeafNATS** (Shi et al., 2019)[2] as our baseline. LeafNATS is an open-source toolkit that can train and evaluate neural Seq2Seq models for the abstractive summarization. The authors modified the PGN by adding an intra-decoder (Paulus et al., 2018) to it.

4.2 Results

CNN/Daily Mail Table 3 shows our main results on the CNN/Daily Mail test set with ROUGE. Underlines indicate statistically significant differences from the baseline using the bootstrap test (Dror et al., 2018). The results for the Lead-3 baseline method are shown at the top, with excellent abstractive representatives in the middle, and our reconstruction methods at the bottom.

From Table 3, we can explicitly observe that the sequence-level reconstruction mechanism beat the baseline but took three times as long as the original model to train (117.9 hours), which is computationally expensive and time-consuming. Therefore, even though the sequence-level reconstruction has

[1] https://github.com/nlpyang/PreSumm
[2] https://github.com/tshi04/LeafNATS

Dataset	Train	Validation	Test	Article length	Summary length
CNN/Daily Mail	287,226	13,368	11,490	781	56
Newsroom	993,101	108,621	108,670	751	30

Table 2: Basic statistics of the datasets.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 Baseline (See et al., 2017)	40.34	17.70	36.57
Pointer-Generator (See et al., 2017)	36.44	15.66	33.42
Pointer-Generator + Coverage (See et al., 2017)	39.53	17.28	36.38
PreSumm (Liu and Lapata, 2019)	41.72	19.39	38.76
Pointer-Generator (our implementation)	36.69	15.90	33.45
+ Sequence-level reconstruction	37.10	16.32	33.92
+ Word embedding reconstruction	38.21	16.85	35.01
+ IDF-weighted embedding reconstruction	38.48	17.05	35.23
Pointer-Generator + Coverage (our implementation)	39.45	17.31	36.01
+ Word embedding reconstruction	40.02	17.91	36.74
+ IDF-weighted embedding reconstruction	40.40	18.21	37.12
PreSumm (rerun)	41.2	18.99	38.29
+ IDF-weighted embedding reconstruction	**41.55**	**19.17**	**38.56**

Table 3: ROUGE F1 scores on the test set of the CNN/Daily Mail dataset. The best results of our experiments are marked in bold. Underlined results significantly surpass the PGN, coverage or PreSumm baseline with $p < 0.01$.

been proved to work well for NMT (Tu et al., 2017), it is not suitable for the summarization task. One of the most likely explanations is that NMT is a sentence or document transformation between two languages, which attempts not to lose any information during the process. However, summarization is designed to compress the information to form a shorter version of the original article. The nature of this task makes it extremely difficult to reproduce the whole input sequence from the generated summary. Furthermore, recovering the entire original information is not necessary and does not match the summarization objective. Therefore, we report the results of the PGN-based sequence-level reconstructor only on this dataset.

For both the PGN and the Coverage baseline, we can see higher ROUGE scores achieved by the word embedding-level reconstructions, which demonstrates the their effectiveness. In addition, our third reconstructor with IDF-weighted embeddings outperformed the baselines and two other reconstruction methods, despite far fewer training epochs.

Even though we did not observe as much improvements as the previous two baselines with the PreSumm-based reconstruction, the statistical significance test shows the stability and effectiveness of our method. To overcome the appearance of rare words, PreSumm tokenize words into subwords with Byte Pair Encoding (Sennrich et al., 2016). However, one subword may appear in words with various IDF values, which makes it meaningless to calculate IDF in the granularity of subwords. Therefore, in our experiments, we still calculated IDF on the word-level while we gave the IDF weights for the words only to their first subword. We believe that the difference of the granularity between IDFs and embeddings is the main reason of the slight improvement. We leave how to solve this issue as our future work.

Newsroom Table 4 lists the comparison results on the Newsroom dataset. Following the previous one, we enumerated extractive methods, abstractive or mixed Seq2Seq models, and our reconstruction architectures in three blocks in turn. Obviously, our reconstruction-based models were largely superior to the extractive methods and two abstractive approaches, while achieving comparable ROUGE scores with ExtConSumm, which is the state-of-

Method	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 Baseline (Grusky et al., 2018)	32.02	21.08	29.59
TLM (Subramanian et al., 2019)	33.24	20.01	29.21
ExtConSumm Extractive (Mendes et al., 2019)	39.40	27.80	36.20
LeafNATS (Shi et al., 2019)	39.91	28.38	36.87
Pointer-Generator (our implementation)	37.12	25.27	33.92
+ Word embedding reconstruction	38.76	26.88	35.47
+ IDF-weighted embedding reconstruction	39.19	27.32	35.95
LeafNATS (rerun)	39.01	27.21	35.77
+ Word embedding reconstruction	39.36	27.49	36.15
+ IDF-weighted embedding reconstruction	**39.57**	**27.79**	**36.34**

Table 4: ROUGE F1 scores on the test set of the Newsroom dataset. The best results from our experiments are marked in bold. Underlined results significantly surpass the PGN or LeafNATS baseline with $p < 0.01$.

Model	Informativeness	Readability	Redundancy
PGN	3.98	4.05	2.90
+ Embed. Rec.	**4.12**	4.12	3.00
+ IDF-wt. Rec.	**4.12**	**4.15**	**3.21**

Table 5: Human evaluation of model-generated summaries.

the-art mixed method. The IDF-weighted word embedding reconstruction was still the best among the reconstructors, which achieved an average improvement of 2.05 ROUGE points over the PGN baseline.

To sum up, these results indicate that simply adding the IDF-weighted embedding-level reconstruction to a Seq2Seq model is a very useful method in abstractive document summarization. However, the training time for introducing the sequence-level reconstructor greatly increased while it gains only a small improvement compared with the other two effective word embedding-level reconstruction methods. We leave the problem of how to successfully reconstruct a long document from a short summary with a neural Seq2Seq model as future work.

5 Analysis

5.1 Human Evaluation

Next, we performed the experiments with a manual evaluation to investigate the quality of summaries in three aspects: informativeness, readability, and redundancy. We randomly selected 100 examples from each dataset. Forty volunteers on Amazon Mechanical Turk (AMT) with a U.S. high school diploma or higher qualification were asked to rate each summary on a scale of 1-5 (higher is better). The average scores for the summaries of each model are shown in Table 5.

As we can see, the baseline model suffered from low informativeness and high redundancy, which can be considered as under-attention and over-attention, respectively. Incorporating the reconstruction architectures could alleviate these problems, whereas better summaries were generated with the help of the IDF values. We consider two reasons for this, as follows. 1) When average word vectors of summaries differ from their source texts, the summaries tend to be penalized by the reconstruction loss. There was no big difference observed between using and not using the IDF weights in terms of informativeness because the word embeddings play an important role in covering all the input tokens. 2) The IDF value for each word serves as a discriminator for avoiding fair attention. The redundancy was reduced with the IDF weights because higher-valued words control the summary content while lower-valued words tend to be ignored.

5.2 Case Study

Table 1[3] shows example summaries obtained by the PGN with and without the IDF-weighted embedding reconstruction. For ease of understanding, we marked four most important passages in the source article with different colors. By observing these example summaries, we identified the following issues: 1) The baseline model tended to incorrectly

[3]Refer back to page 2.

focus on unimportant details in the original text, and 2) The whole sentences or paragraphs were frequently copied from the source even if half of them were meaningless or redundant. For example, the second sentence of the baseline summary indicates that *the cause of the fire accident* has not been investigated. Compared to other elements in the news, e.g., the incident, location, consequences, and solution, the cause is not essential and should not appear in the summary. Moreover, the redundancy can be reduced if the model ignores the attributive clause containing *four alarms* instead of copying the whole sentence from the article.

All of the four key messages were included in the summary generated by our IDF-weighted embedding reconstruction method, while only one of them appeared in the baseline. Even the reference misses one key piece of information. This example demonstrates the efficacy of our reconstruction mechanism in keeping the salient information in text summarization in addition to the improved ROUGE scores.

6 Related Work

Compared to the extractive summarization, abstractive methods are more challenging and attract attention because they can generate new words through the source document representation (Liddy, 2001; Nallapati et al., 2016). With the popularity of deep learning, many neural network-based models, especially Seq2Seq models (Rush et al., 2015; Chopra et al., 2016), have been widely applied to natural language processing tasks, such as machine translation (Bahdanau et al., 2015) and dialogue systems (Lei et al., 2018). Since the work of Rush et al. (2015), neural Seq2Seq networks with an attention mechanism have been widely utilized in the abstractive summarization tasks.

However, the attention mechanism is sometimes not enough to address different problems. For example, repetitions at the word or phrase level cause grammatical errors and insufficient reflection of the main idea of the source article. Therefore, the *distraction* method (Nema et al., 2017) imposes a constraint over the attention that can reduce the probability of repeated content. Tu et al. (2016) and See et al. (2017) found that the original attention often leads to over- or under-focus without the memory of past alignment information. Thus, they used the *coverage* concept from statistical machine translation to keep track of the attention history with an additional loss. Moreover, the inability to handle out-of-vocabulary (OOV) tokens also limits the fluency and readability of generated summaries. To alleviate this problem, hybrid models that combine the extractive and abstractive methods through the copy mechanism account for the vast majority of models used in the summarization task (Vinyals et al., 2015; Gu et al., 2016; See et al., 2017).

Pre-trained language models (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018), essentially word embeddings presenting contextual representations, that were learned from large-scale corpora, have recently emerged and achieved state-of-the-art performances in a variety of NLP tasks (Zhang et al., 2019; Liu and Lapata, 2019; Rothe et al., 2020). Due to the subword tokenizer, out-of-vocabulary words are rarely observed in their output even without the pointer mechanism.

However, neither non-pretrained Seq2Seq models nor BERT can measure the proportion of information transmitted from the source to the target. The idea of *reconstruction* can be implemented in many forms so as to adapt to different types of tasks. For example, Srivastava et al. (2015) proposed an LSTM encoder-decoder model that encodes the video and reconstructs its frame sequence. Tu et al. (2017) proposed a reconstruction model based on NMT that consists of three sequences. If the original sentence can be reconstructed from the target, it proves that the information has been effectively transferred. Another work is SEQ^3 (Baziotis et al., 2019) with a triple sequence structure, which rebuilds the input sentence from the latent representation of the decoder in the unsupervised sentence compression task.

7 Conclusion

In this work, we presented three reconstruction mechanisms for the neural Seq2Seq abstractive summarization task that reconstruct the essential information from the source document to its target summary. The proposed reconstruction methods are applicable to any attention-based Seq2Seq summarization architectures. Experimental results on both the CNN/Daily Mail and Newsroom datasets showed the improvements from the baselines in terms of ROUGE metrics and human evaluation. Our analysis also indicated that the proposed reconstruction approaches can restrict the under-attention to key points and over-attention to redundant parts.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. seq^3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 673–681. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 93–98. The Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4098–4109. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 708–719. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1437–1447. Association for Computational Linguistics.

Elizabeth Liddy. 2001. Advances in automatic text summarization. *Inf. Retr.*, 4(1):82–83.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics.

Afonso Mendes, Shashi Narayan, Sebastiao Miranda, Zita Marinho, André FT Martins, and Shay B Cohen. 2019. Jointly extracting and compressing documents with summary state representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3955–3966. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Preksha Nema, Mitesh Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of*

the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 1063–1072. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Trans. Assoc. Comput. Linguistics*, 8:264–280.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389. The Association for Computational Linguistics.

Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Xiaoyu Shen, Yang Zhao, Hui Su, and Dietrich Klakow. 2019. Improving latent alignment in text summarization by generalizing the pointer generator. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3760–3771. Association for Computational Linguistics.

Tian Shi, Ping Wang, and Chandan K. Reddy. 2019. Leafnats: An open-source toolkit and live demo system for neural abstractive text summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 66–71. Association for Computational Linguistics.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2015. Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 843–852. JMLR.org.

Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *CoRR*, abs/1909.03186.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Zhaopeng Tu, Yang Liu andLifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3097–3103. AAAI Press.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5021–5031. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Conference of*

the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 5059–5069. Association for Computational Linguistics.