Extractive and Abstractive Text Summarization with Reinforcement Learning: Literature Review

Cheng Xi Tsou, Ningyu Chen, and Haotian Zhang

Abstract

In this literature review, we present a high-level summary and a comparative analysis of two papers in the domain of text summarization. The first paper, by Wu and Hu (2018)[15], proposes an extractive summarization model with a focus on semantic and syntactic coherence between extracted sentences. The second paper, by Paulus et al. (2017)[12] proposes an abstractive summarization model that utilizes sequence-to-sequence architectures borrowed from machine translation models (Bahdanau et al. 2014)[1]. Our review provides background information on text summarization and the differences between abstractive and extractive methods. Next, we delve into the first paper by Wu and Hu (2018)[15], where we highlight the novelty of their model and provide a summary of the extractive model architecture. Moving onto the second paper by Paulus et al. (2017)[12], we provide similar discussions and show how the proposed model deals with key issues that came up with abstractive summarization models at the time (Nallapati et al. 2016)[10]. Then, we compare and contrast the objectives, architectures, approaches to solving problems with the models, methodologies, dataset selections, evaluation metrics, and results of the two summarization models. Finally, we explore the potential improvements and future directions in the field of text summarization.

1 Introduction and Background

Text summarization is the process of generating a concise and precise summary of a given text by condensing large amounts of information into shortened versions while retaining all the important points. It's a common task in the field of natural language processing that is applicable in many spheres such as media monitoring, newsletters, financial reports, social media marketing, and much more. There are two main approaches to summarization: Extractive and Abstractive summarization.

Extractive summarization is a more traditional approach to the natural language processing task of text summarization, involving selecting a subset of sentences from an input document to form the summary. Some of the more recent works in extractive summarization have used deep neural networks such as Recurrent Neural Networks by approximating extractive labels to train the RNN (Nallapati et al. 2017)[11], achieving a higher ROUGE score than some abstractive summarization systems (Nallapati et al. 2016)[10].

Abstractive summarization systems generate new phrases, sentences, or words that may have not appeared in the input documents. In a paper published by Nallapati et al. (2016)[10], the authors proposed an attention decoder-encoder Recurrent Neural Network model originally used for machine translation (Bahdanau et al. 2014)[1] for the task of text summarization that outperformed other state-of-the-art summarization models at the time. Since then, there has been a shift in text summarization models from extractive to abstractive systems, where models like BART and BRIO (Lewis et al., 2019; Liu et al., 2022)[7 9] are purely abstractive in nature.

While abstractive summarization dominates most of the state-of-the-art models, there are still key problems with abstractive summaries. Generated summaries don't guarantee readable, coherent, and grammatically correct sentences. When Nallapati et al. (2016)[10] applied their model to the CNN/Daily Mail dataset, the generated summaries were often unnatural and consisted of repeated phrases. There is also the issue of exposure bias when the ground truth is provided at each step of generation during training which is absent during test time, where an error could propagate down the line with one bad generation. Lastly, these systems can only take in short input documents and output even shorter summaries. For example, summaries on the DUC-2004 dataset generated by the state-of-the-art model by Zeng et al. (2016)[16] are capped at 75 characters.

In the first focus paper Learning to Extract Coherent Summary Via Deep Reinforcement Learning (Wu and Hu 2018)[15], the paper avoids these issues by proposing an extractive summarization system with performance on par with state-of-the-art models. The proposed system uses a neural coherence model and a reinforcement learning framework that uses coherence as a reward to improve its extractive summaries.

The second focus paper A Deep Reinforced Model for Abstractive Summarization (Paulus et. al 2017)[12], the paper addresses these issues by building on the sequence-to-sequence model first proposed by Bahdanau et al. (2014)[1] for the machine translation task and introducing a key attention mechanism and a new learning objective that combines maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning.

1.1 ROUGE Evaluation Metric

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is an evaluation metric commonly used for text summarization models. Understanding this can provide further context behind the results from the papers we will discuss. The ROUGE score computes the overlap between the generated summary and reference summary in terms of N-grams, awarding higher scores to summaries with more shared N-grams. A higher ROGUE score suggests that the extracted summary has captured the most important information from the source text. Most text summarization models are evaluated on the ROUGE metric variants ROGUE-1, ROGUE-2, and ROGUE-L, where it counts unigrams, bigrams, and the longest common subsequence-gram respectively.

$$\text{R-}n = \frac{\sum_{s \in \text{reference summary}} \sum_{\text{gram}_n \in s} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{s \in \text{reference summary}} \sum_{\text{gram}_n \in s} \text{Count}(\text{gram}_n)}$$

2 Evolution of Extractive Summarization

In the early days of summarization, the task relied heavily on the use of rule-based algorithms, such as 'importance evaluator' and 'Production rule system for summarization,' as well as other simple statistical methods like 'System using term statistics,' to calculate the importance of sentences in a text corpus. Later on, significant progress was made in the summarization task through the introduction of neural networks. A recurrent neural network was utilized to delete words from sentences for a sentence compression task (Filippova el at. 2015)[4]. Then convolutional neural networks were trained to extract highly ranked sentences into a summary (Cheng and Lapata 2016)[3].

As the field advanced, the summarization task was transformed into a reinforcement learning problem, with models designed to make a sequence of extraction decisions to maximize the ROGUE score of the output. ROGUE score quantifies the similarity between the generated summary and the ground truth summary. However, prior to the research of Wu and Hu (2018)[15], no work has been done to apply coherence as part of the reward function.

Wu and Hu introduces a novel approach that integrates coherence into the reward function, resulting in the extraction of more coherent and readable summaries. Their Reinforced Neural Extractive Summarization (RNES) model combines the ROUGE score with a neural coherence model. This novel approach empowers the model to extract summaries that were not only informative but also coherent.

3 Reinforced Neural Extractive Summarization Model

Wu and Hu adopts a two-step approach to training their RNES model, combining the strength of both supervised learning and reinforcement learning. In the first step, they pre-train an initial Neural Extractive Summarization (NES) model using supervised learning to initialize a robust foundation model that captures some basic sentence and document-level features that are relevant to the summarization task. This initialization process allows the model to converge faster and achieve better performance when fine-tuned with reinforcement learning. In the second step, they employ reinforcement learning to fine-tune the model, introducing coherence and informativeness scores into the reward function through an integration of a neural coherence model and ROGUE score into the function.

3.1 Neural Extractive Summarization Model

Wu and Hu develops an NES model that uses a hierarchical deep neural network architecture with two processing levels: word level and sentence level. The input sentences are represented as a matrix of word embeddings. At the word level, a convolutional neural network (CNN) is employed to extract word features, transforming the sentence matrix into a weighted features vector of its words. At the sentence level, bi-directional gated recurrent units (GRUs) are employed to capture the context of the sentence, generating forward and backward hidden states. These hidden states are concatenated to form the contextual representation of the sentence. The probability of extracting the t-th sentence is then computed by fully connected layers of perceptrons. This probability is conditioned on the contextual representation of the t-th sentence, all sentences extracted before the t-th sentence, and the representation of the entire document, which is computed by a non-linear transformation of the mean of all sentence contextual representations. Wu and Hu trains this NES model using supervised learning with the ground truth extraction labels that are available.

3.2 Reinforcement Learning

After the NES model is pretrained, Wu and Hu further fine-tunes the model using reinforcement learning. The reward function plays a pivotal role in reinforcement learning as it guides the optimization process toward the desired outcome. Wu and Hu designs the reward function to encourage the extraction of coherent and informative summaries. To achieve this, the reward function combines the coherence score and the ROGUE score. The coherence score serves as the immediate reward and is computed through a neural coherence model designed and pre-trained. The ROGUE score is used as the final reward, assessing the overall quality of the extracted summary.

3.3 Neural Coherent Model

Wu and Hu design and pre-train a neural coherent model to compute cross-sentence coherence. The model is based on the Adaptive Recursive Convolutional (ARC-II) model proposed by Hu et al. (2014)[6] for sentence matching tasks. The ARC-II model can effectively capture complex hierarchical structures and matching patterns of sentences, enabling the model to understand the semantic relationship between sentences. Similar to Hu et al.'s (2014)[6] model, the neural coherent model consists of interleaving convolution layers and max-pooling layers, followed by fully connected layers of perceptrons:

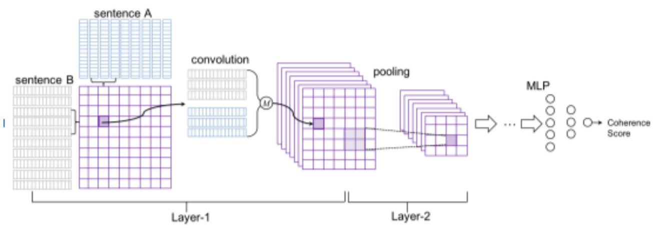

Fig. 1 Neural Coherent Model (Hu et al. 2014)[6]

The model is trained using a pairwise training strategy, which aims to encourage the model to assign higher coherence scores to coherent sentence pairs. By doing so, this model is pre-trained to learn to differentiate between coherent and incoherent sentence pairs. Once the model is trained, it is fixed before integrating into the reinforcement learning framework of the RNES model.

3.4 Reward Function

The reward function can be simply represented as $Reward = ROGUE(G) + \lambda Coherence(G)$, where G denotes the extracted summary, is a coefficient that balances the two rewards. The $Coherence(G)$ is the sum coherence score of G:

$$Coherence(G) = \Sigma_{(S_A,S_b) \in adj(G)} Coh(S_A, B_b)$$

$adj(G)$ is the set of adjacent sentences in G, and the function $Coh(\cdot,\cdot)$ denotes the neural coherent model.

4 Abstractive Encoder Decoder Model

In A Deep Reinforced Model for Abstractive Summarization (Paulus et. al 2017)[12], the paper builds on the encoder-decoder model proposed by Bahdanau et al. (2014)[1] initially used for the machine translation task. To tackle the aforementioned problem of repeated phrases, Paulus et al. (2017)[12] introduce a key attention mechanism: intra-temporal attention in the encoder that takes account of previous attention weights while a sequential intra-attention model in the decoder takes account of previously generated words. Another problem that Paulus et al. aim to reduce is exposure bias, in which the paper proposes an objective function that combines the maximum-likelihood cross-entropy loss and rewards from policy gradient learning. The model also aims to improve readability and generate coherent summaries which previous abstractive models have struggled with and can be overlooked with the ROUGE evaluation metric. The following sections will break the model down into three parts: the new attention mechanisms, token generation, and the new hybrid objective function.

4.1 Encoder-Decoder Model

Before we get into the abstractive model, we must look into some of the previous works it was built upon. The encoder-decoder model was first used for text summarization in Nallapati et al. (2016)[10], inspired by the model proposed in the paper Neural Machine Translation by Jointly Learning to Align and Translate (Bahdanau et al. 2014)[1]. The encoder-decoder model proposed by Bahdanau et al. (2014)[1] uses a bidirectional-RNN for the encoder and an RNN for the decoder. Since the paper is focused on machine translation, the model introduces an alignment model that outputs a weighted context vector based on the alignment scores of the current input in the current context. Intuitively, this can be thought of as an attention mechanism in the decoder. The paper on machine translation provided important insights into using sequence-to-sequence models with attention mechanisms for other generative tasks, namely abstractive summarization.

4.2 Attention Mechanism

The model first reads in the input sequence with a bidirectional LSTM encoder and computes hidden states from the embedding vectors of the input. For the decoder, we use a single LSTM and compute hidden states from embedding vectors of the output.

At each decoding step, the model defines an attention score by combining the hidden states of the decoder and encoder vectors with a bilinear function. We then normalize the attention weights by penalizing input tokens that have obtained high attention scores in the past decoding steps. Nallapati et al. (2016)[10] have demonstrated this intra-temporal attention function can reduce the number of repetitions when attending to long documents. Finally, we compute the normalized attention scores across the inputs and obtain the input context vector for the encoder.

While the intra-temporal attention function deals with the encoded input sequences, the paper introduces a novel intra-decoder attention mechanism that incorporates previously decoded sequences, a new mechanism that did not present in existing encoder-decoder models for abstractive summarization. At each decoding step, the model computes a new decoder context vector by combining the hidden states of the decoder and the hidden state of the previous time step with a bilinear function. The same normalizing process is also applied.

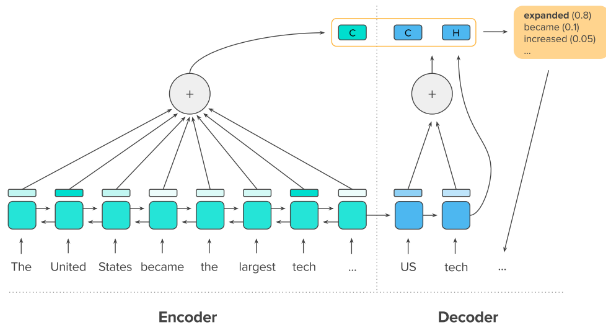

Fig. 2 Abstractive Model Architecture (Paulus et al. 2017)[12]

The figure shows the encoder and decoder attention functions combined and the two context vectors "C" computed from attending over the encoder and decoder hidden states. The two context vectors along with the decoder hidden state "H" are then used to generate a new word. The output is then fed back into the decoder to be used for generating the next word.

4.3 Token Generation

For generating a new token, the decoder uses a switch function that decides at each decoding step whether to use a token-generation softmax layer to generate a new token or a pointer mechanism to copy rare or unseen tokens from the input sequence (Nallapati et al. 2016)[10].

The token-generation layer generates a probability distribution by using a shared weights matrix from the embeddings matrix for the encoder and decoder sequences and a concatenated input of the decoder hidden states and the two context vectors. The practice of a shared weights matrix was used in Press & Wolf (2016)[13] which allowed the token generation function to make use of syntactic and semantic information in the embedding matrix.

The pointer mechanism reuses the temporal attention weights as a probability distribution to copy an input token. During training, the switch function will choose to use the pointer mechanism when the output is an unseen or rare token; otherwise, the token generation layer is used to generate a new token.

Another way that the paper avoids repeated phrases when generating a summary is by keeping track of previously generated trigrams and forcing the decoder to never output the same trigram during beam search.

4.4 Hybrid Learning Objective

The paper aims to solve the problem of exposure bias by proposing a hybrid objective function to train the encoder-decoder model. The most common way to train a decoder RNN for sequence generation is by minimizing the maximum likelihood loss at each step. However, minimizing this loss does not produce the best results when being evaluated on metrics such as ROUGE. The two main reasons why this phenomenon occurs is because the network is provided the ground truth at training time but does not at test time and can cause errors to accumulate as it generates the summary. Secondly, there are many ways to arrange the sentence orders of a potentially valid summary which ROUGE metrics take into account that the maximum-likelihood objective does not.

The paper uses a self-critical policy gradient training algorithm (Rennie et al. 2016)[14] to remedy this issue by maximizing a discrete metric, namely ROUGE. We define a new loss function that uses a reward function that uses the ROUGE metric to compare with the ground truth and two outputs that are sampled and chosen greedily from the probability distribution. While this improves our model's performance on ROUGE, it does not guarantee the increase of readability of the output so we combine the new loss function with the maximum-likelihood loss to form a mixed learning objective.

5 Comparative Analysis

In this section, we will provide a comparative analysis of two papers in the summarization domain: "A Deep Reinforced Model for Abstractive Summarization"[12] and "Learning to Extract Coherent Summary Via Deep Reinforcement Learning"[15]. These papers differ in their objectives, architectures, approaches to solving the problem, training methodologies, dataset selections, preprocessing techniques, metrics, and results.

5.1 Objectives and Approaches

The abstractive model proposed by Paulus et al. (2017)[12] aims to tackle some key issues that are prevalent in the sequence-to-sequence model by Bahdanau et al. (2014)[1] when applied to text summarization. The objectives of the paper are to increase the readability of the generated summary and solve problems specifically in the abstractive summarization domain: reduce repeated phrases and minimize exposure bias at training time. Wu and Hu (2018)[15] go one step further and propose an extractive model to solve the problem of cross-sentence coherency, something that previous summarization models do not consider.

The architectures of both models are different in nature. The abstractive model is a sequence-to-sequence encoder-decoder model with a switch function that chooses an output from either a learned probability distribution or a pointer mechanism that copies from the input. Conversely, the extractive model employs a hierarchical deep neural network architecture, consisting of a CNN at the word level, Bidirectional GRUs hidden units at the sentence level, and followed fully connected layers of perceptrons to output the extraction probabilities for each sentence. Both papers also use reinforcement learning in different ways. Paulus et al. (2017) [12] use a self-critical policy gradient training algorithm (Rennie et al., 2016) [14] just for training the encoder-decoder model whereas, in the extractive model, the reinforcement learning framework is used to fine-tune the neural network and learn a policy that chooses whether to extract a sentence or not.

Both models have similar and different approaches in how they tackled the issue of increasing readability and coherence. Wu and Hu (2018) [15] introduce a mixed reward function that balances between maximizing the ROUGE score and a scaled coherence score in order to improve the model performance on ROUGE metrics while maintaining a coherent output. Similarly, Paulus et al., (2017) [12] introduces a mixed objective function that balances maximizing the ROUGE score through a reward function to decrease exposure bias, and also minimizing maximum likelihood loss to increase the coherency and readability of the output. The interesting thing is when the experiments were conducted, the mixed objective functions in both papers favored maximizing ROUGE metrics. The hyperparameter coefficient λ for weighted coherence score and maximum likelihood loss were $\lambda = 0.01$ and $\lambda = 0.0016$, respectively.

5.2 Dataset and Evaluation Metrics

The dataset that both models were trained on is a modified version of the CNN/Daily Mail dataset (Hermann et al. 2015) [5]. The dataset contains 287,113 training examples, with 11,490 testing examples. Paulus et al. (2017) [12] limit the input lengths to 800 tokens and the output length to 100 tokens while the extractive model limits the input length of each sentence to 50 words. The abstractive model also tests the model on the New York Times dataset, which has more varied but shorter summaries. The NYT dataset is preprocessed so the input and output token lengths are similar to that of the CNN/Daily Mail dataset.

The evaluation metrics that both papers use are ROUGE metrics, namely ROUGE-1, ROUGE-2, and ROUGE-L, and human evaluation. Paulus et al. (2017) [12] use ROUGE-L as part of the reward function to train the encoder-decoder model so we expect the abstractive model to perform better on the specific variant of ROUGE-L. On the other hand, Wu and Hu (2018) [15] use a mix of ROUGE-1, ROUGE-2, and ROUGE-L as the reward for reinforcement learning, so the improvement of the ROUGE score should be about the same for all variants. Both papers also use human evaluation. The extractive model is evaluated by having human evaluators rank pairs of summaries on the criteria of overall quality, informativeness, and coherence. The abstractive model is evaluated by having human evaluators rate the criteria readability and relevance.

5.3 Results

Both papers evaluate their models using the CNN/Daily Mail dataset. The results of Paulus et al. (2017) [12] show that the intra-decoder attention function improved ROGUE scores, surpassing the state-of-the-art model at that time (Nallapati et al., 2017) [11]. The best-performing model on the ROUGE metric in the Paulus et al (2017) [12] is the RL with an intra-attention model yielded a ROUGE-1 of 41.16, ROUGE-2 of 15.75, and ROUGE-L of 37.75. However, this is to be expected as the RL objective in the abstractive model is trained on the ROUGE-L metric. We can see a drastic improvement in ROUGE score when evaluated on the NYT dataset which makes sense as the CNN/Daily Mail dataset inputs are longer, which is a known challenge in the abstractive summarization domain.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 (Nallapati et al., 2017)	39.2	15.7	35.5
SummaRuNNer (Nallapati et al., 2017)	39.6	16.2	35.3
words-lvt2k-temp-att (Nallapati et al., 2016)	35.46	13.30	32.65
ML, no intra-attention	37.86	14.69	34.99
ML, with intra-attention	38.30	14.81	35.49
RL, with intra-attention	**41.16**	15.75	**39.08**
ML+RL, with intra-attention	39.87	**15.82**	36.90

Fig. 3 Quantitative results for abstractive models on the CNN/Daily Mail test dataset (Paulus et al. 2017) [12]

Model	ROUGE-1	ROUGE-2	ROUGE-L
ML, no intra-attention	44.26	27.43	40.41
ML, with intra-attention	43.86	27.10	40.11
RL, no intra-attention	**47.22**	30.51	**43.27**
ML+RL, no intra-attention	47.03	**30.72**	43.10

Fig. 4 Quantitative results for abstractive models on the New York Times test dataset (Paulus et al. 2017) [12]

In Wu and Hu (2018) [15] the RNES with a coherence model is not the best-performing model, whereas the best-performing model is the RNES without a coherence model, achieving ROUGE-1 of 41.25, ROUGE-2 of 18.87, and ROUGE-L of 37.75. Wu and Hu (2018) [15] attribute the lower ROUGE scores of the RNES with a coherence model, with ROUGE-1 of 40.95, ROUGE-2 of 18.64, and ROUGE-L of 37.41, to the reasoning that ROUGE scores do not take into account coherency, it is expected that evaluating RNES with coherence leads to a drop in ROUGE score. Overall, all models from the analyzing papers surpassed the previous state-of-the-art models in terms of ROGUE scores.

Model	R-1	R-2	R-L
Lead-3	39.2	15.7	35.5
(Nallapati et al. 2016)	35.4	13.3	32.6
(Nallapati et al. 2017)	39.6	16.2	35.3
(See et al. 2017)	39.53	17.28	35.38
NES	37.75	17.04	33.92
RNES w/o coherence	**41.25**	**18.87**	**37.75**
RNES w/ coherence	40.95	18.63	37.41

Fig. 5 Quantitative results for extractive models on the New York Times test dataset (Wu and Hu 2018) [15]

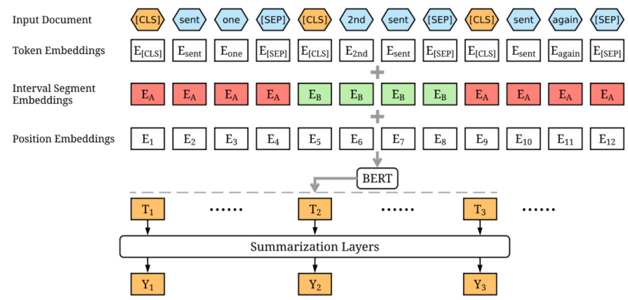

Fig. 6 The overview architecture of the BERTSUM model. (Liu 2019) [8]

The ROUGE score does not fully measure a summary's readability or informativeness. It focuses on N-grams overlap between generated and reference summaries, neglecting coherence and overall readability. High ROUGE scores don't guarantee readability, and summaries capturing essential information with different phrasing may have lower scores. Hence, ROUGE is an imperfect evaluation metric for text summarization.

To address this issue, the authors of both papers conduct human evaluations to ensure the increase or decrease in the ROUGE scores is also followed by an increase or decrease in human readability and quality. The results indicated that the models yielded a more readable and informative summary regardless of the ROUGE scores. While we can normalize these different score ranges and compare them cross-paper, the human evaluation process is too different to draw a meaningful comparison.

6 Conclusion

In this paper, we have provided an overview of extractive and abstractive summarization techniques, discussing the history and state-of-the-art models in the field. We have presented a detailed analysis of the structure and mechanisms of the models presented in the papers, highlighting the unique contributions and benefits over previous models . Finally, we have identified potential future directions for the development of more advanced summarization models.

6.1 Future Developments

The technique of text summarization has continued to develop and evolve since the introduction of the two models we've compared and analyzed in both extractive and abstractive summarization fields.

6.2 Extractive Domain

Although there aren't any improvements done on the specific model of Wu and Hu (2018) [15], the field of extractive summarization has seen rapid development. One such model is BERTSUM, which was published by Liu (2019) [8], an extension of BERT developed for extractive summarization.

BERTSUM creates token embeddings, segment embeddings, and position embeddings from the input document and feeds them into a BERT layer. Once processed with the BERT layer, it generates the predicted score through a summarization layer. It demonstrates strong performances compared to earlier models, as evidenced by its ROGUE scores when tested on the CNN/DailyMail dataset. Drastic improvements of BERTSUM over other state-of-the-art models at the time may indicate that the future of extractive summarization models will likely involve leveraging large-scale pre-trained language models, such as BERT and GPT.

6.3 Abstractive Domain

The paper we chose in the abstractive summarization seems to be more influential in its domain, with several subsequent papers building off of Paulus et al. 's work. An example of such work is Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting by Chen and Bansa (2018) [2].

Chen and Bansal (2018) [2] builds on previous work by Paulus et al. (2017) [12] by solving the persisting issue of slow and inaccurate encoding for long documents. Their proposed model takes advantage of recent developments in both extractive and abstractive domains to create an extractive-abstractive architecture with a policy reinforcement learning framework to bridge the two networks together.

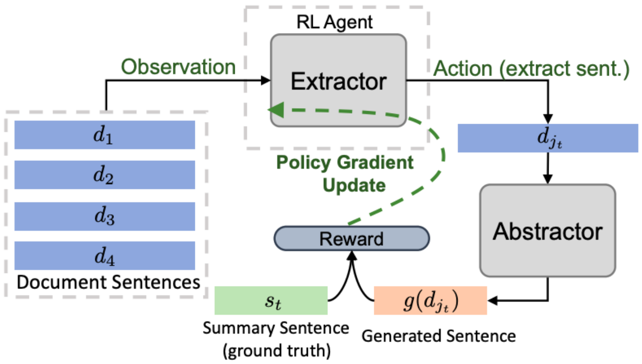

Fig. 7 Reinforced training of the extractor (for one extraction step) and its interaction with the abstractor. (Chen and Bansal 2018)[2]

As shown in the figure, the proposed system uses an extractor that selects important sentences and sends it to an encoder-decoder abstractor network, where it rewrites the extracted sentences.

Key challenges such as encoding long documents, generating repeated phrases, and exposure bias will always be present in the abstractive summarization domain. However, with new ideas to tackle these problems and mitigate bias, we can expect the future of abstractive summarization to combine advantages from previous works to create an even better summarizer.

References

1 Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

2 Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. *CoRR*, abs/1805.11080, 2018.

3 Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *CoRR*, abs/1603.07252, 2016.

4 Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

5 Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

6 Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. *CoRR*, abs/1503.03244, 2015.

7 Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

8 Yang Liu. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318, 2019.

9 Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. Brio: Bringing order to abstractive summarization, 2022.

10 Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016.

11 Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press, 2017.

12 Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304, 2017.

13 Ofir Press and Lior Wolf. Using the output embedding to improve language models. *CoRR*, abs/1608.05859, 2016.

14 Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563, 2016.

15 Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. *CoRR*, abs/1804.07036, 2018.

16 Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. Efficient summarization with read-again and copy mechanism. *CoRR*, abs/1611.03382, 2016.