Literature Review - Document Summarization

Haojun Li
haojun@stanford.edu
Department of Computer Science, Stanford University

1 Introduction/Task Definition

The document summarization task is defined as compressing and summarizing a long article into a short paragraph of several sentences. There are currently 2 main ways of achieving document summarization, extractive and abstractive summary. Extractive summary casts this problem as a binary classification problem where the model aims to predict whether a source sentence will be part of the summary. Predicting the membership of source sentences in summary has achieved great results as most sentences that are part of the summary either directly came from the source document or is a paraphrase of a sentence that is part of the source document with minimal modification. The main problem with extractive summary methods is that it tends to produce sentences that are not coherent. Since they restrict the summary sentences to be directly extracted from the source document, it limits the coherency of the summary. However, this limitation has proven to be beneficial comparing to abstractive summary methods. Abstract summary aims to improve extractive methods in that it not only select the most relevant details from the source sentence, but also generates new tokens that is not present in the source document. Thus, abstractive methods could ”paraphrase” sentences found in the source document and form a more coherent summary. Earlier works on abstractive summary cast this problem as a machine translation problem, where given a source document the model should be able to ”translate” the source document into a succinct summary. However, as multiple author soon realized, this does not necessarily improves extractive methods since the model sometimes generates repeated paraphrasing, and replaces words present in the source document with related but ultimately incorrect phrases. Thus, extractive methods are still achieving better results than any current state-of-the-art abstractive methods.

The most common method of evaluation is the ROGUE scores (ROGUE-1, ROGUE-2, ROGUE-L) where it calculates the unigram/bigram/longest common sequence overlap between the gold summary and the generated summary. However, as with QA and MT task’s BLEU score, this metric does not capture key metrics such as coherency and gramaticallity. Thus, these metrics heavily biases against abstractive methods since abstractive summary tends to produce more ”novel” words in a more coherent manner.

The dataset that is commonly used for this task is the CNN/DM dataset and the NYT dataset, both are very rich datasets. The CNN/DM dataset consists of stories collected from CNN and Daily Mail news articles with several bullet points that ”highlights” the news articles. The NYT dataset consists of stories collected from New York Times with a short summary written by librarians. These dataset also have the problem that most sentences in the summary are copied from the source text with minimum modification, which again biases against abstractive methods.

2 Summary of Articles

In this literature review I will summarize 6 articles, 4 of which are major contributions to abstract text summarization task and the remaining 2 are recent works in extractive summarization tasks. I intend to focus my project on the abstractive summary task while borrowing ideas from extractive summarization tasks.

2.1 Seq-seq model

The first major contribution in abstractive text summarization using neural methods comes from IBM Watson team (Nallapati et al., 2016). Following the advancements in neural machine translation (NMT) systems with sequence-to-sequence models, this paper describes a way to apply these models to the text summarization task with several improvements. The main idea is that they first used an off-the-shelf attentional encoder-decoder RNN, and it turns out that it already out perform the current state-of-the-art systems. This will be used as their baseline model which they will improve on. Details of encoder-decoder RNN models are fairly well known so will not be described here (and it is also not described in any of the papers). They introduced several improvements listed below

1. They restricted the output vocabulary to only contain words present in the source document in an attempt to limit the computational capacity required, and turns out this works fairly well since most words in the summary are present in the source document at some position.

2. They introduced several new embeddings in addition to word embeddings (which they used word2vec). These embeddings corresponds to part-of-speech, named-entity-tags, and TF-IDF statistics of the words. They then concatenated these embeddings into a single embedding for each word before feeding it into the encoder bi-directional LSTM. The decoder uni-directional LSTM will produce words one by one while attending to all source document positions. This allows the decoder to not only take into account the state of each output word conditioned on all previous words, but also attend to all source sentences and use those hidden states to inform the decoder's decision for the next word.

3. To solve the well-known OOV (Out Of Vocabulary) issue, they provided a ”switch” mechanism, which is activated when the model predicts the UNK token, and the generator will calculate a ”pointer” distribution over all source sentences to pick the best word to replace this UNK token with. This allows the model to produce rare words as well as named entities such as names. To train the model, they provide the explicit pointer position when the model predicts an UNK, and the explicit pointer points to the first occurrence of the gold word.

4. They also added hierarchical attention mechanism in which they have attentions at word-level and sentence-level informing the decision of the decoder.

They achieved great results with this model, but it require a lot of computational resources and the summary outputs are not always ideal. They noted that even though the words may appear to have high overlap with the gold summaries, it tends to not be factually correct and it also tends to repeat itself. Several papers after this one attempt to solve these problems with more advanced techniques, but it turns out that none of the papers are able to beat the extractive methods in terms of ROGUE score metrics. Again, the higher ROGUE scores does not necessarily mean a model is better.

2.1.1 Compare and Contrast

This is the founding article in which they introduced several new methods using neural networks for this summary generation task. Most other papers will use the results in this paper as a baseline model. Although the results achieved in this paper is surpassed many times over by later papers, the methods used in this paper is still significantly influential in later papers. They also achieved state-of-the-art results on both datasets.

2.2 Pointer Genreator

A year after the previous paper, a new challenger emerges with a pointer generator network (See et al., 2017). The authors have noted that there are many existing undesirable behaviors such as factually inaccurate details, inability to deal with OOV, and repeating themselves. The authors proposed a new pointer-generator model in which, at each position of the output, they calculate an additional ”generation probability” which is then used to bias the attention distribution of all source positions. This generation probability essentially serves as a soft ”switch” at every position of the output, comparing to the hard switch only at OOV presented in the previous paper. The authors also proposed reusing the attention distribution calculated by the encoder with the generation probability, which also improved the results.

To solve the repetition problem, the authors proposed a coverage mechanism commonly seen in machine translation systems where they calculate a ”coverage loss” aimed to penalize repeated attention to the same locations. This effectively reduced repetitions.

Note that their model took 3 days and 4 hours to train on a Tesla GPU.

2.2.1 Compare and Contrast

Unlike the previous paper, they do not have pre-trained embeddings and they train their embeddings from scratch. The reason why is beyond me. During test time, they used beam search, another common decoding mechanism, to find the best decoding sentences with the highest likelihood. This method alone could have improved previous systems but they did not have ablation studies on how beam search is able to improve results. The results of this model is significantly higher than that presented in the previous paper, but they are still unable to beat the metrics from extractive methods. They noted the reason is due to the inflexibility of ROGUE scores, which, as I have described before, favors extractive methods due to the high overlap between summary sentences and source sentences.

The author also noted that since they could copy from the source text at any position of the output due to their soft switch mechanism presented in the previous paper, their system has a lower degree of abstraction. In particular, their system copies whol article sentences 35% of the time while reference summaries do so only 1.3% of the time. They realized that their model learned to copy instead of generate due to the fact that copying can easily reduce the loss significantly than generating new words.

A new addition to this paper is the unanonymized version of the dataset. Previous papers which operates on the CNN/DM dataset will first use entity extractors to remove all entities from the document and replace them with tags. The author of this paper believes that learning to recognize entities is also an important objective of this task, so they propose working on the unanonymized dataset. They achieved state-of-the-art scores on both datasets.

2.3 Bottom-Up Attention

A year after the previous paper, several students from Harvard (Gehrmann et al., 2018) made significant improvements on the pointer generator model presented above with a simple content selector. They frame the selection problem as a sequence-tagging problem, identifying tokens from a document that are part of its summary. Then, they experimented to either use a hard mask to mask out sentences and words that the content selector deem unimportant according to some tunable threshold, or multiply that probability into the attention distribution to effectively ”bias” the distribution of the pointer generator network to favor words that the content selector deem worthy. This approach can be seen as one of the more successful approaches to combine extractive methods and abstractive methods. Here, they have a content selector that essentially ”extracts” words that seems important and should be present in the summary, while a pointer-generator network then ultimately decides whether the word should be included in the summary.

2.3.1 Compare and Contrast

Unlike the previous paper, the author proposed a 2-step method instead of end-to-end training. The reasoning behind this argument is that when human summarizes texts they also look at what to include first and then paraphrases the sentences that are important, instead of generating sentences as you read through the document. However, the author proposed several end-to-end alternatives in which they jointly train the content selector and pointer-generator network with a soft mask as described above.

The author of this paper also added 2 penalty terms to force the model to produce better results.

1. They added a length penalty term which penalizes shorter sentences and prompt the model to produce longer sentences. They additionally set a minimum length based on the training data.

2. They added a summary-specific coverage penalty different from the penalty term from the previous paper. Again, they do not have ablation studies on how well this change alone could have benefited their model

As with previous paper, they also evaluated on how abstractive the model is. It turns out that the model is mostly extracting sentences and only occasionally generating novel words and paraphrasing. This is in line with the analysis from the previous paper since the ROGUE score heavily biases against abstractive methods that generates new words since gold summaries also copies a lot from the source text.

They achieved state-of-the-art results on these datasets.

2.4 DCA

In a different domain, several advances was made in the deep reinforcement learning areas and several people see it as a way to improve current extractive summary models. During the last 3 years, multiple papers have came out that combines the MLE-based models with reinforcement learning methods to achieve better results through conditioning agents with better reward systems instead of just simply minimizing a loss function. DCA is one such agents (Çelikyilmaz et al., 2018). The key idea of the model is to divide the hard task of encoding a long text into several smaller tasks of summarizing paragraphs by several agents. Then, these agents will "communicate" with each other by sharing their hidden states of their summary with other agents. This communication goes through multiple layers. Essentially, agents are able to summarize their own subsection of the text and "talk" to other agents about their own summarization. This method is similar to the hierarchical attention structure proposed by (Nallapati et al., 2016). Lastly, they will produce hidden states for the decoder to use, and the decoder will attend to the agent attention states. Additionally, they allow the decoder to copy words from different paragraphs by calculating a generation probability similar to (See et al., 2017), but this attention is across multiple agents which attends to the agent hidden states.

The main contribution of this paper is that it formulates this problem as a reinforcement learning problem. Thus, they can better tune the objective function. Instead of simply doing MLE learning, thus turning their model into a hierarchical attention model, they introduced reinforcement learning objectives such as cosine distance between two consecutively generated sentences, reinforcement learning loss to promote exploration vs exploitation, and combining these losses with the MLE loss during training. They also introduced intermediate rewards to promote the agents to produce sentences that maximizes the ROGUE score.

In their experiments, 3 agents performs the best comparing to 2 or 5 agents and they achieved state of the art results on the datasets. Since multiple researchers have realized that ROGUE score does not mean better models, they added human evaluations and achieved much better results from human evaluators.

2.5 Compare and Contrast

Comparing to 3 previous papers, this paper introduces the novel method of "communicating agents" which allows them to formulate this problem as an RL problem. The introduction of RL loss and reward system allows them to produce results that are better than MLE based methods. However, these methods are still unable to achieve better results when comparing to extractive summary methods in terms of ROGUE scores, but they are able to produce better summaries according to "human evaluations".

Another interesting point here is that this method is essentially the same as hierarchical attention structure proposed by multiple papers before, and I believe that those papers can also benefit from reinterpreting their model and cast them as an RL problem. Thus, introducing these RL-loss and reward systems might be able to improve those models as well.

Like the pointer generator model, the authors of this paper also uses beam search, but with a depth of 5 instead of 4. This is an indication that computation power has increased throughout the years that allows these authors to have better models simply because they have access to better computational resources. Thus, the achievements of the models might not be due to the models themselves but rather due to the ability to efficiently train and test them.

2.6 Hierarchical Extractive Model

As described above, even the current state-of-the-art abstract summarization model is unable to beat the extractive summarization models due to the fact that most sentences that are present in the summary are also present in the source document. Thus, it is worth summarizing papers in the extractive summarization space. The current best performing model before BERT (Introduced by a team at Google (Devlin et al., 2018)) is a hierarchical structured self-attentive model (HSSAS) (Al-Sabahi et al., 2018). In this model, they proposed a simple hierarchical structured self-attention mechanism to create the words, sentence, and document embeddings. Then, they cast this problem as a classification task in which the model computes the respective probabilities of sentence-summary membership. This way of framing the problem is standard in extractive document summarization methods.

Sentence-level attention is similar to the abstractive methods described above, in which sentence level hidden states from LSTM or GRU units are used to calculate an attention distribution over all the sentences. These attention states can be interpreted as ”relatedness” between sentences, and thus inform the next layer or the decoder of whether the sentence should be included in the summary or not. LSTM and GRU are notoriously hard in remembering longer-term information. Thus, the author introduced word level self-attention which allows the model to use an weighted average of all previous states as extra input to the recurrent unit, which allows the recurrent unit to effectively ”access” information from several positions away instead of remembering them. The author argued that this is more similar to how human understand document, where we tend to refer back to what we read several positions away in order to write the next word.

2.6.1 Compare and Contrast

The model is not so different from the abstractive models that we have seen so far. For example, the sentence level attention is used in all previous abstractive papers to either calculate a coverage penalty in pointer-generator network (See et al., 2017) or used as during content selection in bottom-up summarization (Gehrmann et al., 2018). However, the main contribution of this paper is that the attended hidden output of the word-level embeddings is used both in the final classification problem as well as input to a sentence-level LSTM recurrent layer. The output of the sentence-level LSTM recurrent layer is then used in the final classification problem as well. Thus, to determine whether the model should select a sentence or not, it will take into account not only the word-attended sentence embedding themselves, but also inter-sentence relations through sentence LSTM. This model has the current state-of-the-art ROGUE scores due to the fact that it is able to correctly identify the sentences that should be in the summary.

One key problem with extractive summary is that the summary written by human are not necessarily extractive for obvious reasons. To solve this problem, the authors have to ”discretize” the summary by generating a gold label for each sentence based on the summary. The most common way of doing this is to greedily maximize the ROGUE score of the ”gold” sentences. Essentially, the authors find the rogue score of all the sentences in the source document against the summary written by humans, and select the top 3 sentences that have the highest ROGUE scores. Then, this problem becomes a simple classification task in identifying those sentences. However, there are obvious limitations to this approach due to the fact that there are inherent differences between paraphrased sentences in the summary vs the source sentences, which abstractive summary is trying (and failing) to solve.

2.7 Fine-tuned BERT

Due to the recent advances with BERT (Devlin et al., 2018), which is a large pretrained contextual embedding using large scale high dimensional LSTM with self-attention, it has dramatically improved many NLP tasks including Machine Translation, Sentiment analysis, and text classification. It is no surprise that this monstrosity is being used in document summarization task as well. Yang is able to modify BERT and produce much better results in extractive summarization than previous authors (Liu, 2019). The major contribution of this short paper is simply modify BERT's architecture to allow training on multiple sentences instead of the 2 sentence default used by BERT to perform a entailment task. Then, the author did fine-tuning with summarization layer using either a simple classifier which consists of a single linear layer with sigmoid output for each sentence, the inter sentence transformer which allow sentences to attend to each other, as well as an additional recurrent neural network that encodes information between sequences of sentences to produce more informed encodings before sending it to a linear layer with sigmoid output.

The author also included several methods used by previous papers, most notably the trigram-blocking, which is a simple way of reducing redundancy by skipping sentences that have a trigram overlapping existing predicted summaries. During inference time, the author calculated the scores of each of the sentences, which are the sigmoid output from the model for each sentence, and picked the top 3 sentences to include in the summary. This is a valid approach since the sigmoid output can be interpreted as the probability that the sentence is included in the summary, and the highest 3 sentences will maximize the likelihood of the sentences being included in the summary.

This paper currently holds the state-of-the-art result in document summarization of the CNN/DM dataset.

2.7.1 Compare and Contrast

This paper shows that with enough computational power (the author used 3 GPUs) and a strong enough pretrained contexual embeddings (BERT is trained for couple days on several TPUs), any basic model can outperform existing baseline models. However, this paper provides a very strong baseline to compare against other models which uses BERT in place of their existing contextual embeddings. Several previous papers either used GloVe or Word2Vec embeddings, which can be swapped out to use BERT embeddings and improve their results. However, doing so also requires huge computational resources and GPUs which are pretty expensive and time-consuming.

3 Future Work

The document summarization task is far from being solved. Here are proposed future works that I'm considering working on.

1. Since extractive methods perform so well, we could have a 2-step method to first select the sentences using ideas proposed by (Al-Sabahi et al., 2018). The content selector proposed by (Gehrmann et al., 2018) can also be used in this case, but with a weaker result. Then, we can treat the selected sentences to the abstract summary as a translation task, forming coherent summaries from the selected sentences.

2. There are several paper proposing sentence enhancements and using no neural networks at all in performing summarization tasks. They mostly rely on Entity extraction and coreference resolution to replace pronouns and references with its original entity. This can be used to augment the extractive methods.

3. Since BERT is so popular, we can also modify BERT and put several networks we have summarized earlier on top of BERT and observe results, but that would be less interesting since most of the work would be engineering efforts and waiting for training.

References

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Asli Çelikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *CoRR*, abs/1803.10357.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. *CoRR*, abs/1808.10792.

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

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.