Experimental Protocol - Document Summarization

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

1 Hypothesis

Following the literature review, I hypothesize that by redefining the abstractive summarization task as a token level extractive summarization task, we would achieve better results boosted by recent advancements in pretrained contextual embeddings. Specifically, I define the summarization task as a token level tagging problem that is different from that defined by the Bottom-Up paper (Gehrmann et al., 2018), and using Bert in a novel way different from the BertSum paper (Liu, 2019). Essentially, I believe that all the tokens needed to generate a coherent summary is in the source document itself scattered across many sentences, and by carefully selecting and rearranging tokens, I should be able to achieve good results.

2 Data

Following See et all (See et al., 2017), I will use the unanonymized CNN/Daily Mail dataset. This dataset consists of total of 312060 (truncated to be a multiple of 10) source document and target summary pairs. The source document is a single news article from these two sources [1] and the target summary is a 3 sentence highlight written by humans. Concatenating these 3 sentences gives us a summary target. I split up the data following BertSum (Liu, 2019) at 90/5/5 train, dev, and test split. Thus, my dataset consists of 280854 training examples and 15603 development and test examples each.

I have already preprocessed the data and tokenized them according to Bert's specification (adding [CLS] tokens and [SEP] tokens, etc.). Since the task was originally an abstractive summarization task, I redefine it as a token level tagging task by scanning through the target summary from start to finish, finding the longest common sequence at each step, and tagging the sequence in the source that is closest to the previously tagged common sequence. Bottom-up paper (Gehrmann et al., 2018) used a similar tagging strategy where they also scanned the target but tagged the first occurrence of the longest common sequence in the source document. I hypothesize that this dramatically reduces model performance since tokens chosen in this fashion will be far away in the source document while close together in the target summary. Not only does LSTM have historically bad performance in longer sequence tasks, this style of tagging forces the underlying LSTM to reference tokens that are far away from each other when deciding whether a token will be included in the gold summary.

3 Metrics

The metrics for automatic evaluation is the same as those done by all previous papers, namely the ROUGE metrics. These metrics are used to measure the unigram recall (ROUGE-1), bigram recall (ROUGE-2), as well as longest common sequence recall (ROUGE-L). As noted by previous authors, there is significant issues with these metrics since, much like BLEU scores for translation tasks, higher metric values does not necessarily mean a better model. Thus, if time and resource permits, I will add some human evaluations and qualitative results as well.

Since I redefine the abstractive task as an extractive task, my ”gold labels” would not allow me to achieve perfect ROUGE scores. After tokenizing the source and target documents according to BERT's specification, the ”gold labels” allows us to achieve 0.8 ROUGE-1, 0.7 ROUGE-2, and 0.8 ROUGE-L scores. This is higher than reported by previous papers because BERT's tokenization mechanism is different than previous

[1] Daily Mail is not really a reliable news source...

papers (which uses Stanford NLP's tokenization scheme). Thus, comparing metrics with previous papers must be done with care. Nonetheless, these scores are really really good, and is also the theoretical limit of my model performance. These will be my oracle scores (a term that is used in all previous document summarization papers).

4 Models

The models for this task is rather simple. There are actually 2 tasks that I'm experimenting with.

1. The first task is a purely sequence tagging task, where we will select words from the source document in the order as they appear in the source document (i.e. finding words to include in the summary). Thus, this task maximizes unigram recall, and hopefully will be useful in later tasks. This task will have BERT as the contextual embedding encoder and I will layer on top:

   (a) A single linear layer mapping the embedding at each position to a real value, which will then pass through sigmoid to find the probability of this token appearing in the summary. We will train on binary cross entropy loss, and find the right threshold during evaluation to find the best cutoff point. This will be my baseline model.

   (b) A transformer layer between BERT and output layer, which allows output units to attend to positions in the source document.

   (c) A bi-directional multi-layer LSTM between BERT and transformer layer to allow even more complexity of the model.

   These models are inspired by BertSum (Liu, 2019), which uses similar layers for sentence-selection style extractive summary.

2. The second task is to not only select the words but also arrange them. I do not intend to complete this task since I do not have much time left and most of my current efforts are spent on task 1, but I'll define it here nonetheless. The task not only seeks to extract words that would be in the summary, but also seeks to arrange them in a way that will be coherent, thus maximizing ROUGE-2 and ROUGE-L. The model would be to have BERT as a contextual embedding extractor, and feed these into a decoder that attends to these embeddings at each decoding step to generate a coherent selection of tokens from the source document. This model will be similar to the Seq-seq models proposed by (Nallapati et al., 2016)

5 General Reasoning

I believe that abstractive summaries that generates novel words as in (See et al., 2017), (Nallapati et al., 2016), and (Çelikyilmaz et al., 2018) are overrated, and I hypothesis that a coherent summary can be generated solely by carefully selecting tokens and arranging them in the right order. The source document is so rich in vocabulary that we should not need to seek to paraphrase them with words not in the source document. In the mean time, we also should move past pure sentence-selection style extractive methods since they greatly constrain us to have exactly 3 whole sentences. This dataset and models that I have described is a perfect test for this hypothesis, such that if I was able to achieve better results than previous abstractive models, it would greatly strengthen this hypothesis. If I was unable to perform well on this task then we can analyze why the summary cannot be generated from the source document vocabulary alone, and these insights will inform later contributors to this task.

6 Progress So Far

I have implemented (with great pain) the entire experimental framework. I preprocessed the data and have defined the models and training code in PyTorch. I have done so by borrowing code from public Github repos such as the BertSum repo[2] and Pointer-Generator repo [3] for preprocessing, and previous course projects for training and modeling. I have gained preliminary results by training the baseline model, and it turns out that it is only able to achieve 0.38 ROUGE-1 score and 0.11 ROUGE-2 score, which is slightly lower than that reported by previous papers. I will seek to tune the model better with a better threshold and hyper parameters.

I don't believe I would get to task 2, but I have implemented the model for task 2 as well. The main constraint at the moment is training time and

[2] https://github.com/nlpyang/BertSum

[3] https://github.com/abisee/pointer-generator

cloud credits, as training with Bert might take a long time. Nonetheless, I will keep training and tuning my models

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.