

Natural language processing

Andrea Auletta Aulo

May 23, 2024

Contents

1	Text summarization with pretrained encoders	2
1.1	Some definitions	2
1.2	Summary	2
1.2.1	Extractive summarization	3
1.2.2	Abstractive summarization	3
1.3	implementation	3
2	Automatic Text Summarization: a State-of-the-art Review	4
2.1	Extractive methods	4
2.2	Abstractive methods	4
3	A Comprehensive Survey on Process-Oriented Automatic text summarization with exploraiton of LLM-based methods	5
3.1	Summarization model	5
3.1.1	Extractive summarization	5
3.1.2	Abstractive summarization	5

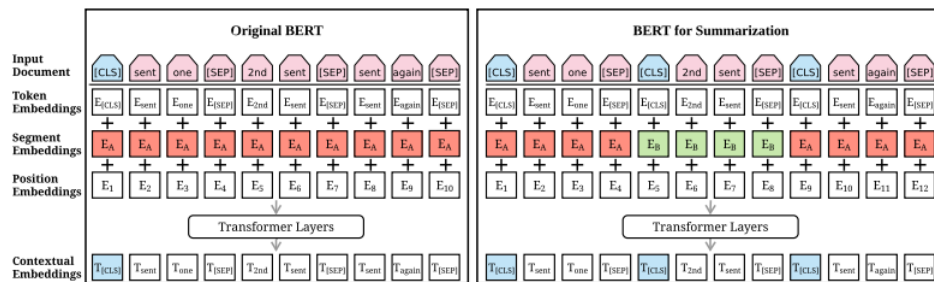
1 Text summarization with pretrained encoders

1.1 Some definitions

- **Good summary:** A good summary must be fluent and consistent, capture all the important topics, but not contain repetitions of the same information;
- **Abstractive modeling:** the task requires language generation capabilities in order to create summaries containing novel words and phrases not featured in the source text;
- **Extractive summarization:** is often defined as a binary classification task with labels indicating whether a text span (typically a sentence) should be included in the summary;
- **Pretrained language model:** extends the idea of word embeddings by learning representations from large-scale corpora using a language modeling objective.

1.2 Summary

Here they explore the potential of Bert under a general framework encompassing both extractive and abstractive summarization. They combine the Pretrained Bert with a randomly-initialized Transformer decoder. The difference here is that we want to manipulate multi-sentential input w.r.t. the usual task of Bert. In Bert for summarization the document representations are learned hierarchically where lower transformer layers represent adjacent sentences, while higher layers (+self-attention) represent multi-sentence discourse.



1.2.1 Extractive summarization

With BertSum we have the vector v_i of the i-th [CLS] symbol from the top layer can be used as the representation of the i-th sentence. After this we have other inter-sentence transformer layers to capture document-level features for extracting summaries. The output layer is a sigmoid classifier.

1.2.2 Abstractive summarization

Standard encoder-decoder framework is used. The encoder is BertSum and the decoder is a 6-layered Transformer initialized randomly. To circumvent the fact that the decoder is not pretrained is designed a new fine-tuning method: Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the encoder and the decoder, each with different warmup-step and learning rates:

- $lr_\epsilon = \tilde{lr}_\epsilon \cdot \min(step^{-0.5}, step \cdot warmup_\epsilon^{-1.5})$, where $\tilde{lr}_\epsilon = 2e^{-3}$, $warmup_\epsilon = 20000$ for the encoder;
- $lr_{\mathcal{D}} = \tilde{lr}_{\mathcal{D}} \cdot \min(step^{-0.5}, step \cdot warmup_{\mathcal{D}}^{-1.5})$, where $\tilde{lr}_{\mathcal{D}} = 0.1$, $warmup_{\mathcal{D}} = 10000$ for the decoder;

The encoder can be trained with more accurate gradients when the decoder is becoming stable.

1.3 implementation

- PyTorch;
- OpenNMT;
- bert-based-uncased: <https://git.io/fhbJQ>;
- dropout for abstractive models;
- rouge-2 score for extractive models against gold summary (select the top 3 sentences);
- Summarization quality using Rouge (1, 2, L)
- human evaluation.

2 Automatic Text Summarization: a State-of-the-art Review

2.1 Extractive methods

The main task is to determine which sentences are important and should be included in the summary.

- Cheng: data-driven approach for single-document summarization based on continuous a hierchical document encoder and an attention-based extractor. The model can be trained on large-scale datasets and learn informativeness features based on continuous representations without access to linguistic annotations. The labels are assigned to each sentence in the document individually based on their semantic correspondence with the gold summary;
- Nallapati: recurrent neural network based sequence model for extractive single-document summarization (SummaRuNNer). Identify the set of sentences which collectively collectively give the highest ROUGE with respect to the gold summary
- Narayan: similar approach but for sentence ranking there is a combination of maximum-likelihood cross-entropy with rewards from policy gradient reinforcement learning to directly optimize the final evaluation metric (ROUGE), a sentence gets a high rank for summary selection if it often occurs in high scoring summaries.
- Yasunaga: GCN takes sentence embedding from RNN as input node feature and through multiple layers-wise propagation generates high-level hidden features for sentences. Sentence salience is then estimated through a regression on top and the important sentences are extracted in a greedy manner while avoiding redundancy.

2.2 Abstractive methods

Here we talk about sequence-to-sequence models (first introduces as enocder-decoder).

- Rush 2015: neural attention based model that generates each word of the summary conditioned on the input sentence. Easy scalable for training on large amounts of data;
- Chopra 2016: extension of the previous model with RNN, the encoder encodes the position of the input words and uses convolutional network to encode input words;
- Nallapati 2016, Switching generator/pointer: decoder is equipped with a switch that decides between generating a word based on the context or using a word from the input.

3 A Comprehensive Survey on Process-Oriented Automatic text summarization with exploration of LLM-based methods

Dialogue summarization: Zero-shot approach: employ discourse relations to structure the conversation and leverage an existing document summarization model to craft the final summary.

3.1 Summarization model

3.1.1 Extractive summarization

- Words-counting model: define importance as most frequent or most likely to occur. TextRank: phrase graph representation, adjacency matrix;
- Similarity-based model: assign high rank based on similarity;
- Classifiers: cut and represent the sentences, categorize the sentences using a classifier select the sentences whose category is summary. Squeeze-BERTSum.

3.1.2 Abstractive summarization

- RNN-based models: RNN, LSTM and GRU - convert the input sequence into a sequence of letters, words or sentences using seq2seq. SummaRuNNer;

- Transformer-based models: BART, GPT.