LongSumm 2021: Session based automatic summarization model for scientific document

Senci Ying, Yanzhao Zheng, Wuhe Zou
NetEase, China
{yingsenci, zhengyanzhao, zouwuhe}@corp.netease.com

Abstract

Most summarization task focuses on generating relatively short summaries. Such a length constraint might not be appropriate when summarizing scientific work. The LongSumm task needs participants generate long summary for scientific document. This task usual can be solved by language model. But an important problem is that model like BERT is limit to memory, and can not deal with a long input like a document. Also generate a long output is hard. In this paper, we propose a session based automatic summarization model (SBAS) which using a session and ensemble mechanism to generate long summary. And our model achieves the best performance in the LongSumm task.

1 Introduction

Most of the document summarization tasks focus on generate a short summary that keeps the core idea of the original document. For long scientific papers, a short abstract is not long enough to cover all the salient information. Researchers often summarize scientific articles by writing a blog, which requires specialized knowledge and a deep understanding of the scientific domain. The LongSumm, a shared task of SDP 2021(https://sdproc.org/2021/sharedtasks.html), opts to leverage blog posts created by researchers that summarize scientific articles and extractive summaries based on video talks from associated conferences(Lev et al., 2019) to address the problem mentioned above.

Most of the previous methods divide the document according to section, and use the extraction or abstraction model to predict the summary for each part respectively, and combine the results as the final summary of the document. Section based method may drop some important information among the sections. Generally, only uses one type of model for prediction can not make good use of the advantages of different models. Combined with the later models and solutions, we propose an ensemble method based on session like figure1.

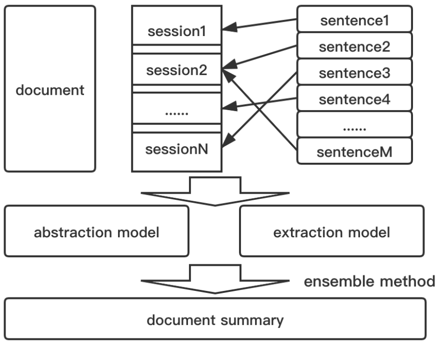

Figure 1: SBAS: a session based automatic summarization model

We split the task into four steps: session generation, extraction, abstraction, and merging the results at the end. First, we split an document into several sessions with a certain size, and use a rouge metric to match the ground truth (sentences from given document's summary). Then, we train two different types of model. One is the abstraction-based model. Specifically, we use the BIGBIRD(Zaheer et al., 2020), a sparse attention mechanism that reduces this quadratic dependency to linear, and PEGASUS(Zhang et al., 2020), a pretrained model specially designed for summarization. The other one is based on extraction method. We test the performance of TextRank(Mihalcea and Tarau, 2004; Xu et al., 2019), DGCNN(Dilate Gated Convolutional Neural Network)(Su, 2018) and BERTSUMM(Liu, 2019). In the end, for each type of model, we generate the summary from the one which has the best performance, and use an ensemble method to merge the summaries together. The result show that our method is effective and beats the state-of-art models in this task.

Proceedings of the Second Workshop on Scholarly Document Processing, pages 97–102
June 10, 2021. ©2021 Association for Computational Linguistics

2 Related Work

The common automatic summarization is mainly divided into the extraction-based summarization and the abstraction-based summarization. The extraction-based model extracts several sentences and words from the original article by the semantic analysis and sentence importance analysis to form the abstract of the article. Typical models include TextRank(Mihalcea and Tarau, 2004; Xu et al., 2019) algorithm which based on sentence importance and the extraction method based on pre-training model(Liu, 2019). The abstracts obtained by the extraction model can better reflect the focus of the article, but because the extracted sentences are scattered in different parts of the article, the coherence of the abstracts is a problem to be challenged. The abstraction-based models are based on the structure of seq2seq, and the pre-training model is used to achieve better generation effect like BART(Lewis et al., 2019), T5(Raffel et al., 2019). Recently, PEGASUS(Zhang et al., 2020), a pre-training model released by Google, specially designed the pre-training mode for the summarization task, and achieved the state-of-art performance on all 12 downstream datasets.

This task focuses on the the solution of the long summary. The input and ouput text of the traditional model is limited due to the memory and time-consuming. However, this task requires the model to summarize scientific papers and generate very long summaries. To solve this problem, most of the solutions in the the previous are based on sections(Li et al., 2020; Roy et al., 2020). They divide scientific papers into sections, generate abstracts for each seciton, and finally combine them to get the final results. Resently, Google's new model BIGBIRD(Zaheer et al., 2020) , using sparse attention mechanism to enable the model fit long text, is suitable for this task scenario.

3 Method

The pre-training model plays a significant role in the field of automatic summarization, but due to its huge amount of parameters, most of the models can only be used for short text tasks. For long articles, there are two common ways to do. One is to directly truncate the long articles, the other is to predict the articles according to the section. This paper proposes a text segmentation method based on session, and use an ensemble method with the extraction model and the abstraction model to generate the final summary.

3.1 Session Generation

Limited by the computational power, many methods chose to truncate long articles directly, which makes the model unable to perceive the content of the following articles, and the generated summary can only reflect part of the input text. Others divide the article into sections, but this also raise some problems. The length and content of section are different between different articles. The division based on section may not reflect the relationship between text and abstract well. This paper proposes a segmentation method based on session, which divides the article into different sessions according to the selected size, predicts the summary for each session, and selects the most appropriate window size in this task by adjusting the size of the session.

The specific data processing steps are as follows: (1) First, select the appropriate session size(2048 words) and a buffer(128 words), which is used to keep the last text of the previous session as the context of the current session. (2) For generating models. The real summary is divided into sentences, and the corresponding summary sentence is assigned to each session according to the rouge metric. In order to make the model predict long summaries as much as possible, a greedy matching rule is used to allocate the summary sentences to each session. we first drop the sentences with the threshold 0.7, which denotes the rouge score between the session and summary sentences. Then we pick the sentences according to the scores until meets the length we set, default 256 words.

Although this may cause different sessions to predict the same summary, we think that duplicate sentences can be detected through the later data processing, and it is more important for the training model to generate long sentences . (3) For the extraction model, we only need to match different sessions with their corresponding summary sentences.

3.2 Abstraction-based Model

The training data contains around 700 abstractive summaries that come from different domains of CS including ML, NLP, AI, vision, storage, etc. And the abstractive summaries are blog posts created by NLP and ML researchers. The traditional generation model is mainly based on the classical transformers structure. In order to solve the problem of long text input , we use the sparse attention structure BIGBIRD(Zaheer et al., 2020), which is proposed by Google recently, and makes finetuning on its two open source pre-training models:

(1) Roberta(Liu et al., 2019): a bert model with the dynamic masking and drops the next predict loss

(2) PEGASUS(Zhang et al., 2020): a transformer model while using gap sentences generation to pre-training.

The models used in this paper are both pretrained on arXiv datasets, so they have strong ability to generate abstracts.

3.3 Extraction-based Model

The extractive data have 1705 extractive summaries which are based on video talks from associated conferences(Lev et al., 2019). We have tried tree different extraction models to select important sentence from the documents.

(1) TextRank(Mihalcea and Tarau, 2004): We simply use the TextRank algorithm to pick out some most important sentences from the documents and limited the number of sentences extracted.

(2) DGCNN-Extraction(Su, 2018): DGCNN is an 1D-CNN Network structure combines two new convolution structure: dilated convolution(Gehring et al., 2017) and gated convolution(Dauphin et al., 2017).

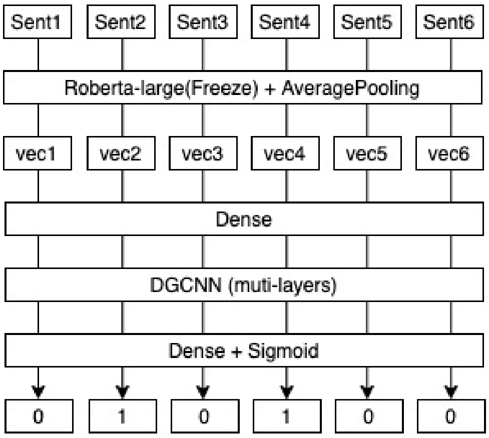

Figure 2: DGCNN-Extraction model structure

The advantage of DGCNN-Extraction model is that it can process the information of every sentence in the text at same time, and identify the important sentence by context. The way we train the model is as follows:

1. We use NLTK to break the original paper into multiple sentences, and label each sentence according to the golden extractive summarize.
2. Transform each sentence by Roberta-Large pre-trained model(Liu et al., 2019), and get the output of last hidden layers as the feature representation, then convert the feature matrix to a fixed-size vector by average-pooling.
3. TRAINING: Feed the obtained sentence vectors into the DGCNN-Extraction model (Figure 2) and binary classify each sentence.
4. INFERENCE: Take the sigmoid-output of the model as the importance score for each sentence, according to which we extract the corresponding sentences from the paper as the extractive summary and the total length of the summary is limited.

(3) BERTSUMM(Liu, 2019): BERTSUMM is a Bert-based model designed for the extractive summarization task. Different from DGCNN-Extraction model, because of the limit of the input length of Bert, we have to divide each paper into sections, then treat each section as a independent sample. As the result, we get 17720 sections in total. Follow the practice in BERTSUMM paper, we insert a [CLS] token before each sentence and a [SEP] token after each sentence and the [CLS] is used as a symbol to aggregate features from one sentence. In each sections we label the [CLS] token of sentences in ground-truth as 1 and others as 0. We split the data into training data and validation data and train the model on the training data. It's a pity that the F1-score of the result of validation data only peaked at 0.35. We think it is because this approach abandon the the information between the sections and the assumption of sections independence is not valid.

According to the performance of this three models on the validation set, we choose DGCNN-Extraction model as the baseline of the extraction model.

3.4 Ensemble Method

Abstraction model and extraction model have their own advantages and disadvantages. The advantage of abstraction model is that it can produce different expression from the original text, and can better summarize the original text, also the generated summary will be more fluent than the extracted summary. However, the disadvantage of this model is that the generated content can not be controlled, and it can not guarantee that the model can predict all the key points of the original text. The extraction model can capture most of the important information directly from the score of the original sentence. Therefore, this paper considers an ensemble method to reorganize the abstracts predicted by the abstraction model and the extraction model so as to further improve the accuracy of the abstracts. The specific implementation method is as follows:

1. self drop: since there are overlapping texts between sessions, the results predicted by the model may have repeated text. This paper first divides the predicted summary into sentences, and judges the sentence similarity according to the rouge metric. The sentences whose similarity are greater than a certain threshold(rough1-f + rough2-f > 0.8) will be determined as repeated sentences, and the longest one (we think that the long sentence carries more information) is selected as the most representative sentence, the rest are dropped.

2. sentence reorder: reorder the abstracted and extracted sentences according to the session. For each session we will predict summaries by both abstracted and extracted model. And we ordered them look like this : $sess_1 : abs_{11}, .., abs_{1n_1}, ext_{11}, .., ext_{1m_1}; sess_2 : abs_{21}, .., abs_{2n_2}, ext_{21}, .., ext_{2m_2}; ..; sess_m$. Because the abstraction model predicts the sentence that is usually a summary sentence, we put it before the extracted sentence in the same session.

3. recall: we will filter the combined summaries again and recall the most useful sentence for the final result. To do this, we used TextRank algorithm and dropped the sentences which scores are under 0.9.

After these steps, the predictions from the different models are well cleaned and merged. The most important sentences are selected from the candidate summaries to form the final result. The experiment shows that the comparison of single model and ensemble method has a significant effect.

4 Experiment

We extract the text from the PDF of paper by using Science Parse(https://github.com/allenai/science-parse). There is a lot of dirty text in the data, which will make the model hard to converge during training. So we clean the text as follows: (1) replace the URL link in the text with [url](2) remove special characters from the text and keep only some common symbols. (3) merge the broken words and remove some words that is not in the word list.

We spilt the text of each paper into sessions,and the best session size by testing should be 1024 words. The buffer size is 128 which we think is enought to keep the context. Each sentence of ground truth is set as the target summary of one of the sessions according to the location of the most similar sentence in the original paper. We use the NLTK to count words of the session. As for pretrained model, all input session are truncated to a maximum of 1024 words, and their target summary are truncated to a maximum of 128 words. Based on the test results, the best generation model is built as follows: The model is fine-tuned on the pegasus-arxiv pre-trained model released by Google which has about 570 million parameters for 20 epochs with a learning rate of 2e-5. The batch size is 8 and the model is trained on four v100(32G) GPUs for about 20 hours. As for building DGCNN-Extraction model, all input papers are truncated to a maximum of 400 sentences(1024d) and 7 DGCNN-layers (with 1,2,4,8,16,1,1 dilation rate) are added to the model. Then we compile the model with Adam optimizer(learning rate = 0.001). The model is trained for 20 epochs on training set and the batch size is set to 32. DGCNN is a lightweight model that only takes 30 minutes to train.

Follow the method mentioned above, we ensemble the summaries obtained from the best generation model and extraction model.

5 Result

We test three different models on the test set: (1)$SBAS_{extract}$: the model only include the DGCNN-extraction model for summary. (2)$SBAS_{abstract}$: the one using the PEGASUS as a base abstractive model to generate the summary. (3)$SBAS_{ensemble}$: the ensemble model of the $SBAS_{extract}$ and the $SBAS_{abstract}$. We compare the final test scores of all metrics with other teams on the leaderboard in Table 1.

The result show that both $SBAS_{abstract}$ and $SBAS_{extract}$ model are competitive. As for the result of $SBAS_{abstract}$, its recall-score is much

Method	$rouge1_f$	$rouge1_r$	$rouge2_f$	$rouge2_r$	$rougeL_f$	$rougeL_r$
$BART$	0.1921	0.1122	0.0533	0.0310	0.1062	0.0620
$Sroberta$	0.4621	0.4377	0.1280	0.1212	0.1701	0.1610
$Sharingan$	0.5031	0.5164	0.1706	0.1744	0.2114	0.2162
$Summaformers$	0.4938	0.4390	0.1686	0.2498	0.2138	0.1898
$CNLP-NITS$	0.5096	0.5234	0.1538	0.1581	0.1951	0.2008
MTP	0.4858	0.4919	0.1330	0.1348	0.1697	0.1714
$SBAS_{abstract}$	0.5080	0.4755	0.1740	0.1634	0.2156	0.2016
$SBAS_{extract}$	0.5275	0.5415	0.1711	0.1747	0.2209	0.2262
$SBAS_{ensemble}$	**0.5507**	**0.5660**	**0.1945**	**0.1998**	**0.2295**	**0.2357**

Table 1: Result for Long Scientific Document Summarization 2021

lower than F1-score, this might be caused by the summary generated by $SBAS_{abstract}$ is shorter than the ground truth. We limit the length of summary extracted by $SBAS_{extract}$ to 900 words, and get an excellent result compared with other teams. The result of $SBAS_{ensemble}$ is far superior to the others models, we believe this is because our ensemble method not only remove the redundant sentences in the combined summary, but also make the output of $SBAS_{extract}$ well supplement for the result of $SBAS_{abstract}$.

We extract some of the abstract for manual evaluation, and find that the abstract generated by our method can generate sentences with high readability and cover a lot of important information of the paper, but sentence to sentence is not coherent, the fluency of the abstract is insufficient. And we will try to improve the fluency of the summary in future work.

6 Conclusion

Pre-train models such as Bert and GPT have obvious effects in all NLP fields, but they can't deal with long text due to their huge amount of parameters and computation. In this paper, we propose an ensemble model based on session for the LongSumm task. In our method, the document is firstly segmented according to the session, and some context semantics are reserved. Then, the labels corresponding to each session are matched by a specific algorithm to generate a new dataset. The extraction and abstraction models are trained on the new dataset, and the final summary is obtained by merging the results of different models through the ensemble method. The method proposed in this paper considers the context of the text as much as possible while limiting the memory growth, so that the summary predicted by the model is more coherent. And the method of merging two different types of summary models is proposed for the first time. The prediction results of different models are dropped and combined for the second time, so as to make the results closer to the real summary.

Our model has achieved the best performance in all metrics of this task, but there for improvement. The current approach is to compress the input and output to make the task adapt to the model, but the best design idea is to make the model fit the task. One of the biggest problems is how to reduce the resource consumption of the transformers structure model. BIGBIRD model proposed by Google alleviates this problem through sparse attention mechanism, but after our test, because of the decoding part of the model still uses full attention, BigBird does not solve the problem of long text output, and it is difficult to directly generate a complete long summary from scientific documents in this task. Therefore, future research can focus on how to decode longer text, so that the language model can adapt to more NLP scenarios.

References

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. *arXiv preprint arXiv:1906.01351*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. 2020. Cist@ cl-scisumm 2020, longsumm 2020: Automatic scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2020. Scientific document summarization for laysumm'20 and longsumm'20.

Jianlin Su. 2018. Dgcnn: a reading comprehension model based on cnn.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive text summarization. *arXiv preprint arXiv:1910.14142*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.