The concept of system for automated scientific literature reviews generation

Anton Teslyuk

National Research Center "Kurchatov Institute", Moscow, Russia
Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region

Abstract. We present a concept of system which is aimed to create a literature review of scientific articles having a small sketch of statements as the input. Key elements of the system include transformer-based BERT encoder, deep LSTM decoder and a loss function which combines auto-encoder loss and forces generated summaries to be in the input text domain. We propose to use PMC open access subset for model learning.

Keywords: text summarization · auto-encoder · NLP · BERT · LSTM

1 Introduction

Recent advances in machine learning methods demonstrate impressive results in a wide range of areas including generation of a new content. State of the art methods techniques based on generative adversarial networks (GANs), variational auto-encoders (VAE) and autoregressive models allow to generate images, videos, voice, texts which are very close or even indistinguishable from those create by humans. The success of such algorithms is determined by the availability of large structured datasets which allow to train complex models. Among promising sources of structured data which can be used for development of generative models one can distinguish scientific papers. A number of large paper collections are available including arXiv.org [1] and Pubmed Central [18]. Paper texts are well structured, labeled with keywords, summarized by abstract and title and are organized in a citation network. Such data seems to be very attractive source of information to train sophisticated algorithms for text processing and generation.

In this paper we present a design of the algorithm which will help to create a literature review based on a draft sketch. The idea is correlated with sketch to image and text to image synthesis algorithms, when the model learns to generate photo realistic images having simple sketch or text description as an input. Our model learns to generate blocks of text with a summary about some topic guided by a simple description of the topic as the input. The model includes encoder to map text sequences to latent space, decoder to do the reverse mapping and a loss function which shapes the latent space and forces representations of text blocks about similar topics to be grouped together.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

2 Related research

The task of creation new literature review based on a small sketch is a special case of text summarization task when large text or collection of texts are to be compressed in a more compact summary containing most important points [3]. There are two main approaches to do text summarization: extractive when the most important sentences are extracted from original text and abstractive when a new summary content is created.

Extractive summarization is a classification problem, the input text is split into sentences and each sentence is classified either to be selected for summary or not. It can be trained in either supervised mode when both source text and extracted summary is available in input data or in unsupervised when only source text is used. A common approach for this type of problems is to construct a mapping of sentences to some vector space and then use general purpose classifier to select important sentences. The mapping from sentences to vectors can be done using manually constructed features like TF-IDF [5], convolutional neural network [24], recurrent neural network [14], recurrent network with attention mechanism [23]. Current state of the art methods use transformer-based models [22] for sentence encoding with a stack of self-attention layers [7], [10] which significantly outperforms previous methods. Extractive summarization can be done in unsupervised mode. In this case sentences are also mapped into some vector space and a clustering algorithm is applied to select sentences which are nearest to cluster centers [17]. Each cluster is supposed to contain sentences about some distinct topic. Finally a single sentence from each cluster is included in summary.

Abstractive summarization problem is a special case of sequence to sequence learning problem. A common approach for this sort of problems is to train encoder-decoder model which will encode input text into a vector or a sequence of vectors in latent space and then decode it into smaller sequence. Sutsveker [20] proposed to use deep recurrent neural networks (RNN) for a similar problem in machine translation where input sequence in one language needs to be translated into output sequence in target language. Later RNN approach was significantly improved by introducing attention mechanism [2], [13] which enriches sequence representation by the use of information from intermediate hidden states of recurrent neural network. Vaswani [22] extended this idea in transformer model by using only a deep stack of self-attention layers without using recurrent networks in encoder and decoder. These methods were successfully applied for abstractive summarization problem including recurrent networks with attention [15] and transformer model [11].

The peculiarity of the summation task is that we need to transform multiple sequences from the input into a single summary sequence in the output. For this task a similar encoder-decoder model is used. The main difference is that the output of the encoder is not sent to the decoder directly. Rather one can use sampling from distribution in latent space inferred by encoding input sequences. One can encode a number of text sequences to be summarized into latent space vectors, then do some averaging to sample latent representation of summary

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

and then decode summary latent space vector into a summary text. There are several approaches to do the sampling and training the encoder-decoder. Liu [9] proposed to use generative adversarial networks (GANs) having LSTM encoder, attention-based LSTM decoder which generate summaries for input text and a convolutional network as descriminator to classify synthetic summaries from those generated by human. Although GANs are often used to sample from latent space Liu only used it to train encoder-decoder pair, not for sampling. Analysis of features of latent space constructed by GAN is an interesting topic for further research.

Another approach to construct latent space is to use variational auto-encoders (VAEs) which impose additional regularization constraints for latent space. Latent space representation constructed by VAE have important feature for sampling: near points in latent space lead to similar or "meaningfull" sequences in data space. Shumann [19] demonstrated applicability of VAE for abstractive summarization task. A similar approach was presented by Chu [6] who trained encoder-decoder pair with a combination of auto-encoder loss forcing decoder output to be similar encoder input and similarity loss which forces representation of averaged summary in latent space to be close to latent vectors of original texts.

3 Input Data

To build the model we have used collection of scientific papers from Pubmed Central Open Access Subset collection [18]. This dataset includes more than 643k full-text papers from biomedical and life science journals with total volume of 75.5 Gb. Papers from this collection follow Pubmed Central Tagging guidelines which introduce a rich set of metadata and a standardized structure. The full texts are available in a structured form in XML format.

From the paper, we extract citations and a text content around every citation: N-sentences before and after it. In this way we obtain a number of text blocks containing content related to a paper being cited. Later when constructing the encoder from text sequences to latent space we will use this information to reduce distance between latent vectors for text content related to the same article. This idea can be further improved if we use additional information from paper citation network as a similarity measure in encoder latent space. One can use graph node similarity between paper nodes in citation network [12] as a distance factor between text blocks representations in latent space or use neural network mapping from citation graph space to latent space, e.g. graph2vec [16]. This is a topic for our further research.

We used Pubmed parser library [21] to extract paragraphs and reference entities from XML files. Then we applied NLTK toolkit [4] to split paragraphs into sentences and selected up to five sentences around every citation (two before and two after the citation). In this way we obtained dataset containing more than 12M text blocks labeled with citations of 643k papers.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

4 Model Description

The general scheme of our model in presented in Figure 1. It includes transformer encoder (BERT pre-trained model), recurrent network decoder, averaging of text block representations with the same label in latent space and a feedback from summary decoder to minimize distance between text blocks with the same label and their summary in latent space.

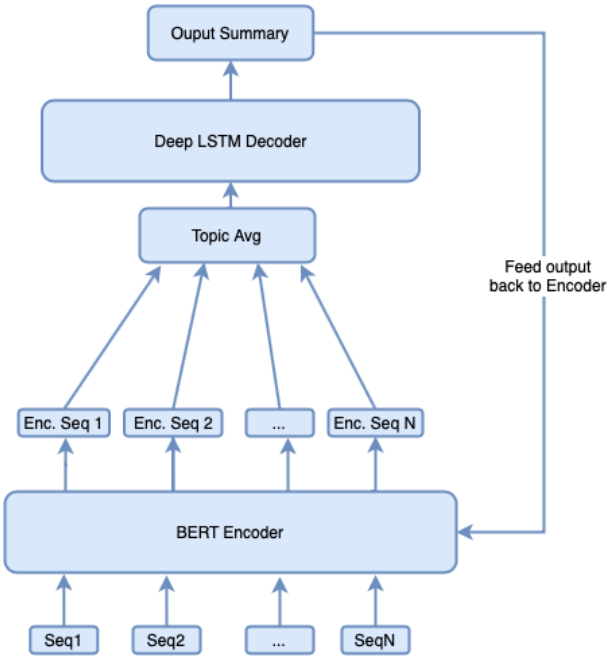

Fig. 1. Model scheme. A set of sentences on the same topic is passed through encoder and then is averaged to calculate topic representation in latent space. After that LSTM decoder in used to generate topic summary. The summary sequence is passed through encoder again to estimate the distance in latent space between summary representation and averaged topic representation. This distance is added to model loss function.

The input is a labeled sequence of text blocks, each text block includes up to five sentences around citation and a label with ID of a paper being cited. Then each text block is encoded using transformed-based BERT encoder [7]. We use BioBERT [8] version of BERT, pre-trained using biomedical text corpora which shows better results for life science specific texts. BERT output is a set of context-aware representations of every token in input sequences. Token representation is a 768-dimension vector. For further processing a set of token representations have to be squeezed into a single vector representation of the whole sequence. To do this a number of methods are proposed: using single

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

token representation, average and max pooling of all tokens, hierarchical and attentive pooling. We use representation of the first <CLS> token, indicating start of the sequence as a representation of the whole sequence. This method is used in original BERT model for next sequence prediction task, when two sequences are fed into BERT encoder and the task is to classify if they are neighbors in the text. During training BERT encoder is fine-tuned to produce embedding of the first <CLS> token optimized for use as a hidden initial state in the first layer of LSTM decoder.

After encoding we average representations of sequences corresponding to the same label. In this way we obtain averaged representations of summaries to be created. This requires us to put enough examples of every label in the batch. In every batch we stack 16 text blocks of two labels, having total batch size of 32 and two summary vectors in the output.

Then averaged representations are fed into recurrent decoder. As a decoder we use stack of four LSTM layers, each having 768 hidden units and a fully connected layer of vocabulary size with softmax activation which outputs probabilities for output words. We initialize hidden state of the first LSTM layer with the output sequence from encoder. The rest LSTM layers hidden states are initialized using transformations of encoder output with fully connected layer with RELU activation. In this way each LSTM layer receives its own optimized initialization vector.

To compute loss we use the idea of Chu [6]. The loss is a sum of auto-encoder loss, which trains encoder and decoder to learn effective representation of sequences in latent space and the averaged similarity between encoder output for generated summary and encoder output for original texts. The idea can be further improved if use additional discriminator classifier which learns to distinguish generated summary output from auto-encoder output of initial text blocks.

When the model is trained it can be used to generate summaries. A brief sentence describing the topic is sent to the model input. Then we calculate encoder output for the input sentence and use K-nearest neighbors algorithm to get nearest K vectors in latent space from training data. Then set of K-vectors from training dataset is averaged and the mean vector is sent to decoder to generate output sequence for the summary.

5 Summary

In this paper we presented a general concept of a system to create a literature reviews. The system is based on abstractive text summarization methods: auto-encoder which combines transformer BERT encoder with LSTM decoder and additional loss factor which shapes latent space to be suitable for summary representation sampling. We believe that having focused to a particular area of life science literature make the task easier than trying to build a general purpose summarization system. Additional advantage is the availability of a big corpora of research papers in PMC collection which are well structured and enhanced

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

with rich metadata. Currently the system is under our intensive development and testing.

Acknowledgements. This work has been supported by NRC Kurchatov institute project "Development of modular platform for scientific data processing and mining" (Project No. 1571).

References

1. arxiv.org e-print archive. https://arxiv.org/
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Basiron, H., Jaya Kumar, Y., Ong, S.G., Ngo, H.C., C Suppiah, P.: A review on automatic text summarization approaches. Journal of Computer Science **12**, 178–190 (2016)
4. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
5. Christian, H., Agus, M.P., Suhartono, D.: Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). ComTech: Computer, Mathematics and Engineering Applications **7**(4), 285–294 (2016)
6. Chu, E., Liu, P.J.: Meansum: a neural model for unsupervised multi-document abstractive summarization. arXiv preprint arXiv:1810.05739 (2018)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics (09 2019). https://doi.org/10.1093/bioinformatics/btz682, https://doi.org/10.1093/bioinformatics/btz682
9. Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., Li, H.: Generative adversarial network for abstractive text summarization. In: Thirty-second AAAI conference on artificial intelligence (2018)
10. Liu, Y.: Fine-tune bert for extractive summarization. arXiv preprint arXiv:1903.10318 (2019)
11. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345 (2019)
12. Lu, W., Janssen, J., Milios, E., Japkowicz, N., Zhang, Y.: Node similarity in the citation graph. Knowledge and Information Systems **11**(1), 105–129 (2007)
13. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
14. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
15. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
16. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005 (2017)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32

17. Nomoto, T., Matsumoto, Y.: A new approach to unsupervised text summarization. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 26–34 (2001)
18. Pmc open access subset. https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
19. Schumann, R.: Unsupervised abstractive sentence summarization using length controlled variational autoencoder. arXiv preprint arXiv:1809.05233 (2018)
20. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
21. Titipat, A., Acuna, D.: Pubmed parser: A python parser for pubmed open-access xml subset and medline xml dataset (2015). https://doi.org/10.5281/zenodo.159504, http://github.com/titipata/pubmed_parser
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Yu, H., Yue, C., Wang, C.: News article summarization with attention-based deep recurrent neural networks. Tech. rep., Stanford University, Tech. Rep
24. Zhang, Y., Er, M.J., Zhao, R., Pratama, M.: Multiview convolutional neural networks for multidocument extractive summarization. IEEE transactions on cybernetics **47**(10), 3230–3242 (2016)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_32