BE SPECIFIC : ABSTRACTION BASED SUMMARIZATION

[1]Adarsh Gupta, [2]Afreen Sultana, [3]Ayush Singh Tomar, [4]Digvijay Khanapurkar, [5]Ms. Visalini
[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professors
[1,2,3,4,5] Information Science and Engineering
[1,2,3,4,5] The Oxford College of Engineering, Bangalore, India

Abstract : Text summarization can be used by personal or specialised assistants. Apart from that it can be used for many personalized devices or applications like mail clients, report generation, news feed etc. There is a problem of searching for relevant documents from the number of documents available, and observing relevant information from it. Earlier there was a technique called Extraction in which summary of document was generated by the words which is already present in the document, no new words were added. But ABS uses a new method called Abstraction which adds new words to form a summary. A long text needs to be summarized, so that human can understand it easily. A method called novel sentence-level policy gradient used by ABS to bridge the non differentiable computation between two neural networks in a hierarchical way. ABS first selects salient sentences and then rewrites them abstractively to generate a concise overall summary. To generate a concise overall summary, they are not restricted to simply selecting and rearranging passages from the original text. Using CNN/Daily Mail dataset ABS achieves the way to find for significantly higher abstractiveness scores. It first operates at the sentence-level and then the word-level. ABS enables parallel decoding of neural generative model that results in substantially faster inference speed.

IndexTerms **- Abstraction. Extraction, Text Summarization**

I.INTRODUCTION

Extractive and Abstractive are two main paradigms of document summarization. The former method directly chooses and outputs the salient sentences (or phrases) in the original document. The latter abstractive approach involves rewriting the summary, and has seen substantial recent gains due to neural sequence-to-sequence models Abstractive models can be more concise by performing generation from scratch, but they suffer from slow and inaccurate encoding of very long documents, with the attention model being required to look at all encoded words (in long paragraphs) for decoding the generated summary word (one by one sequentially). When generating multi-sentence summary, Abstractive models also suffer from redundancy (repetitions). Thus, our method incorporates the abstractive paradigm's advantage

To address both these issues and combine the advantages of both paradigms, we propose a hybrid extractive-abstractive architecture, with policy-based reinforcement learning (RL) to bridge together the two networks. Similar to how humans summarize long documents, our model first uses an extractor agent to select salient sentences or highlights, and then employs an abstractor network to rewrite (i.e., compress and paraphrase) each of these extracted sentences. To overcome the non-differentiable behavior of our extractor and train on available document-summary pairs without saliency label, we next use actor critic policy gradient with sentence-level metric rewards to connect these two neural networks and to learn sentence saliency. We also avoid common language fluency issues by preventing the policy gradients from affecting the abstractive summarizer's word-level training, which is supported by our human evaluation study. Our sentence-level reinforcement learning takes into account the word-sentence hierarchy, which better models the language structure and makes parallelization possible. Our extractor combines reinforcement learning and pointer networks, attempt to solve the Traveling Salesman Problem. Our abstractor is a simple encoder-aligner-decoder model (with copying) and is trained on pseudo document-summary sentence pairs obtained via simple automatic matching criteria.

This method incorporates the abstractive paradigm's advantages of concisely rewriting sentences and generating novel words from the full vocabulary. In addition, we surpass the popular lead-3 baseline on all ROUGE score it adopts intermediate extractive behavior to improve the overall model's quality, speed, and stability. Instead of encoding and attending to every word in the long input document sequentially, our model adopts a human-inspired coarse-to-fine approach that first extracts all the salient sentences and then decodes (rewrites) them (in parallel). This also avoids almost all redundancy issues because the model has already chosen non-redundant salient sentences to abstractively summarize (but adding an optional final reranker component does give additional gains by removing the fewer across-sentence repetitions).

Empirically, our approach is the new state-of the-art on all ROUGE metrics as well as on METEOR of the CNN/Daily Mail dataset, achieving statistically significant improvements over previous models that use complex long-encoder, copy, and coverage mechanisms. In addition, ABS surpass the popular lead-3 baseline on all ROUGE scores with an abstractive model. Moreover, our sentence-level abstractive rewriting module also produces substantially more novel N-grams that are not seen in the input document, as compared to the strong flat-structured model. This empirically justifies that our RL-guided extractor has learned sentence saliency, rather than benefiting from simply copying longer sentences. We also show that our model maintains the same level of fluency as a conventional RNN-based model because the reward does not leak to our abstractor's word-level training.

Overall, our contribution is we propose a novel sentence-level RL technique for the well-known task of abstractive summarization, effectively utilizing the word-then-sentence hierarchical structure without annotated matching.

II.PROBLEM STATEMENT

With the large amount of data that is present today, there is an increased need for summarization of data. It is faster to read a summary of any article rather than reading the whole article. Summarization, as done by humans, involves analyzing and understanding of an article, website, or document to find the key points. For humans, summary generation is easy and straight forward but it involves lots of time. As of now there exists applications like inShorts which uses humans to summarize news into an article of at most 60 words. We plan on improving this by automating the summarization.

PROPOSED SYSTEM

So we have proposed a system by which Text Document can be easily summarized, we consider the task of summarizing a given long text document into several (ordered) highlights, which are then combined to form a multi-sentence summary. Formally, given a training set of document-summary pairs $\{x_i , y_i\}_{i=1}^{N}$, our goal is to approximate function f . This latent function f can be seen as an extractor that chooses the salient (ordered) sentences in a given document for the abstracting function g to rewrite. The overall ABS model consist of two main modules, the extractor agent and the abstractor network.

III. RESEARCH METHODOLOGY

3.1 Modules Description

Preprocessing: In this The Tokenized stories are read from file, lowercased and written to serialized binary files **train.bin**, **val.bin** and **test.bin**. and it will be placed in the newly created finished_files directory. Additionally, a **vocab** file is created from the training data. This is also placed in finished_files

Word Embeddings: Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation, a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.

Extraction Labels: Automatic keyword extraction is process of selecting phrases and words from the **text** document that can at best project the core sentiment of the document without any human intervention depending on the model.

Train Abstractor: The network is trained as an usual sequence-to-sequence model to minimize the cross-entropy loss of the decoder language model at each generation step.

Evaluation: Evaluate standard ROUGE1, ROUGE-2, and ROUGE-L on full length F_1 (with stemming) following previous works.

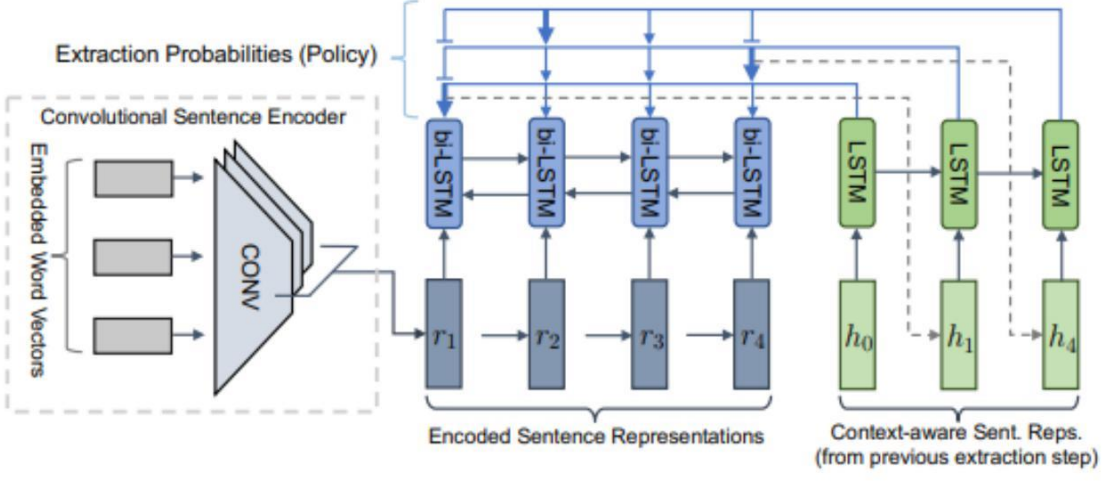

Figure 1: Extractor Agent

Preprocessing Steps:

1. Input CNN and Daily Mail data path.
2. Check the size of file.
3. Create cnn_tokenize, dm_tokenize and finished_files folders.
4. Tokenize CNN and DM data and store in respective folders.
5. The Tokenized stories are read from file, lowercased and written to tar files train.tar, val.tar and test.tar inside finished_files folder.
6. Additionally, a vocab file is created in .pkl format from the training data.

Word Embeddings:

1. Create a list of all the train files.
2. Create an object for Gensim. model = gensim.models.Word2Vec(size=args.dim, min_count=5, workers=16, sg=1)
3. Pass the list of sentences.
4. Train the model on all the train data.
5. Save the output in .bin as well as .w2v format.

Extraction Labels:

1. Extraction of only train and validation dataset to learn patterns.
2. Read each stories from respective folders.
3. Categorise and tokenize the article and abstract from each story.
4. Calculate the recall value.
5. Write the max recall value and line number, for each highlights in the story.

Train Abstractor:

1. Gradient techniques are applied to optimize the whole model.
2. To make the extractor an RL agent, we can formulate a Markov Decision Process (MDP).
3. Denote the trainable parameters of the extractor agent by $\theta = \{\theta_a, \omega\}$ for the decoder and hierarchical encoder respectively. then train the extractor with policy-based RL.

Evaluation:

1. RL training phase, we add another set of trainable parameters v_{EOE} (EOE stands for 'End-Of-Extraction') with the same dimension as the sentence representation.
2. The pointer-network decoder treats v_{EOE} as one of the extraction candidates and hence naturally results in a stop action in the stochastic policy.
3. Set the reward for the agent performing EOE to ROUGE-1 whereas for any extraneous, unwanted extraction step, the agent receives zero reward. The model is therefore encouraged to extract when there are still remaining ground-truth summary sentences.

IV. Result and Discussion

Evaluated our models with the standard ROUGE metric, reporting the F1 scores for ROUGE1, ROUGE-2 and ROUGE-L (which respectively measure the word-overlap, bigram-overlap, and longest common sequence between the reference summary and the summary to be evaluated). Also report the lead-3 baseline (which uses the first three sentences of the article as a summary), and compare to the only existing abstractive and extractive models on the full dataset. given that the disparity in the lead-3 scores is (+1.1 ROUGE-1, +2.0 ROUGE-2, +1.1 ROUGEL) points respectively, and our best model scores exceed by (+4.07 ROUGE1, +3.98 ROUGE-2, +3.73 ROUGE-L) points, estimated that to outperform the only previous abstractive system by at least 2 ROUGE points all round. ABS model with coverage improves the ROUGE and METEOR scores further, convincingly surpassing the best abstractive model. Despite the brevity of the coverage training phase (about 1% of the total training time), the repetition problem is almost completely eliminated, which can be seen both qualitatively and quantitatively.

Producing the summary, ABS simply concatenate the extracted sentences from the extractors. From Table 1 and Table 2, we can see that our feed-forward extractor out-performs the lead-3 baseline, empirically showing that our hierarchical sentence encoding model is capable of extracting salient sentences. The reinforced extractor performs the best, because of the ability to get the summary-level reward and the reduced train-test mismatch of feeding the previous extraction decision. The improvement over lead-3 is consistent across both tables, showing that our pointer-network extractor and reward formulations are very effective when combined with A2C RL.

Validation set

Models	ROUGEs (R-1, R-2, R-L)	METEOR
ABS	(41.23, 18.45, 38.71)	21.14

Test set

Models	ROUGEs (R-1, R-2, R-L)	METEOR
ABS	(40.41, 17.92, 37.87)	21.13

Figure 2: Result

V. Conclusion

ABS propose a novel sentence-level RL model for abstractive summarization, which makes the model aware of the word-sentence hierarchy. Applied our model to a new and challenging long text dataset, and significantly outperformed the abstraction. ABS model train and test is performed on both CNN/DM versions for different stories, along with a significant speed-up in training and decoding.

VI. Acknowledgment

We thank the anonymous reviewers for their helpful comments. This work was possible because of Google collab for providing GPU to train our model.

REFERENCES

[1] Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 209–220. Association for Computational Linguistics.

[2] Sebastian Henß, Margot Mieskes, and Iryna Gurevych. 2015. A reinforcement learning approach for adaptive single- and multi-document summarization. In International Conference of the German Society for Computational Linguistics and Language Technology (GSCL-2015), pages 3–12.

[3]Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.