INTERPRETABLE AND TABLE-STRUCTURED ABSTRACTIVE TEXT SUMMARIZATION

Ren Zhu
Northeastern University
charleszhuren@gmail.com

Yinuo Liu
Zhejiang University
zjuonellllyn@gmail.com

Xusheng Yang
Central China Normal University
yxs160902@163.com

January 9, 2022

ABSTRACT

Existing generative text summaries pay little attention to semantic information in sentences. Therefore, we scored and sorted the information in the original text to select the most meaningful sentences and extract critical information, then structured the representation of the data in a table format, and linearized the table to a string to convert a table-to-text generation problem into a text-to-text generation problem. Finally, we fine-tuned T5-small by using the preprocessed string and the raw document as the input. This approach has been proven to have an excellent performance on the Xsum dataset and is beneficial for improving interpretability and controllability in text summary generation.

1 Introduction

In today's world, we are exposed to a huge amount of information every day. Therefore, summarization is important and necessary. It can help shorten the time needed for reading, fasten the search for specific information, and get the most amount of information on one topic (Abualigah et al., 2020).

Text summarization is a task to shorten a text document while maintaining the salient information of the original text (Jadhav & Rajan, 2018) and can be broadly classified into extractive and abstractive summarization. Extractive summarization selects parts of the source text to make a summary, while abstractive summarization generates summaries with some new words and expressions.

Abstractive text summarization is recently catching attention rapidly and has become an active area of research. It can provide more informative and concise summaries than extractive models. In this work, we propose a novel approach for the abstractive summarization task by representing the source texts in a structured way and performing text-to-text summarization, thus allowing for better interpretability in the generation process.

1.1 Related Work

Abstractive Text Summarization. In the recent past, sequence-to-sequence models (Bahdanau et al., 2014), which are deep-learning models that map an input sequence into another output sequence, have been used for abstractive text summarization tasks (Rush et al.2015). See et al. (2017) introduced the pointer network, which handles out-of-vocabulary issues and has also gained popularity. More recently, large-scale pre-trained models have significantly improved summarization performance (Liu & Lapata, 2019).

Evolved from Seq2Seq models by using attention mechanism but without relying on RNNs, Transformers (Vaswani et al., 2017) gained much popularity. Previous models based on Transformers, such as BERT and GPT, did an excellent job in a wide variety of text summarization tasks. These summarization models have high word overlap performance. However, they tend to copy long passages of the source text, so the summaries might not be quite abstractive (Kryscinski et al., 2018).

More recently, Lewis et al. (2019) introduced BART, a denoising autoencoder built with a Seq2Seq transformer model which pre-trains a model combining bidirectional and auto-regressive transformers and works effectively when fine-tuned for text generation. Zhang et al. (2020) proposed pre-training large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective and thus introduced the PEGASUS model. Raffel et al. (2019) released the T5 (Text-to-Text Transfer Transformer) model, which consists of a transformer-based encoder-decoder architecture and performs well in text-to-text generation tasks.

The state-of-the-art abstractive text summarization models are mostly based on BART or BERT. For example, GLM (General Language Model), a large-scale pre-training model, uses the Transformer architecture similar to BERT (Du et al., 2021). And the R-Drop takes BART as the backbone and fine-tunes it using regularized dropout (Liang et al., 2021). They all achieve good performances.

However, in human language, sentences usually involve various structures, while all these models encode sentences with little consideration for modeling multiple semantic structures (Zhang et al., 2020).

Structured Representations. Researchers have begun to introduce structured representations into text summarization in recent years. For example, Opinosis (Ganesan et al., 2010) first constructs a textual graph, each node representing a word unit with directed edges representing the structure of sentences, and then uses properties of this graph to explore and score various subpaths that help in generating candidate summaries. Another graph-based framework EdgeSumm (El-Kassas et al., 2020), first constructs a new text graph model representation from the input document and then searches and selects the valuable sentences.

Yet, both models only select certain subgraphs that are meaningful and representative to form the final summary instead of generating new words and expressions, thus being more extractive than abstractive.

Closer to our work, very recent work has proposed abstractive summarization while considering the text structure. Frermann and Klementiev (2019) introduced neural models for abstractive, aspect-driven document summarization that can induce latent document structure to identify aspect-relevant segments of the input document. A more relevant model was StructSum, proposed by Balachandran et al. in 2021. This model augments the encoder-decoder summarization model with rich structure-aware document representations based on implicitly learned and externally-derived linguistic structures. However, both do not directly build the structured representations to do the encoding. Instead, they add an attention mechanism to the encoder to augment the sentence representations with sentence dependency information.

Data-to-Text Models. Some text generation model generate descriptive text from structured data such as graphs and tables. Some focus on introducing structure into the encoder. For example, the Graph Transformer uses explicit relation encoding and allows direct communication between two distant nodes (Dwivedi & Bresson, 2012); the hierarchical encoder uses a two-level architecture, first encoding entities and then the data structure (Rebuffel et al., 2020). Others focus on innovating decoding modules based on planning and templates, like the two-step decoder proposed by Puduppully et al. in 2018.

A novel way to do a data-to-text task is proposed by Kale and Rastogi (2021). They cast the data-to-text task into the text-to-text framework by representing the structured data as a flat string and got a great result in data-to-text generation by pretraining and fine-tuning T5 model.

1.2 Contributions

In this work, we propose a new framework to perform abstractive text summarization, which is the first method in this field that builds structured table representations of the input and allows to better interpret and control the output of summarization models by checking those representations. We first extract the structured information from original texts and store them in table format. Then the table is transformed into a string using a linearization algorithm. Finally, we perform text-to-text generalization from the string to generate the final summary.

Our main contribution is threefold:

- First, we propose a novel approach for the abstractive summarization task. Precisely, we extract critical information and store it in the form of a table, and then linearize the table into strings and perform text-to-text summarization from this linearized structured representation.
- Second, we demonstrate that this method allows for better interpretability and control in the generation process by inspecting the structured input representation.
- Finally, we run experiments to fine-tune T5-small in our proposed method on XSum for text summarization. The result shows that it outperforms robust baselines for all ROUGE scores.

2 Background

Named Entity Recognition. Named Entity Recognition (NER) identifies entities, which is one of the vital information in the text. Previous work has already provided several excellent NER software like SpaCy, which can assign labels to spans of tokens.

Coreference resolution. Coreference resolution aims at solving the problem of finding all expressions that correspond to the same entity in the text (Alexey K. Kovalev et al., 2021). Early research applied cutting-edge deep-learning and reinforcement-learning techniques to it (Wiseman et al., 2016; Clark & Manning, 2016), while recent works have favored neural approaches (Thomas Wolf, 2017) like Huggingface coreference resolution model (NeuralCoref).

Sentence-level sentiment analysis. It's the task to analyze each sentence's sentiment (positive/negative/neuter) in the whole text. A pre-trained huggingface model, Google's T5-base fine-tuned on IMDB dataset, plays a good job using transfer learning and a unified text-to-text transformer.

Part of Speech (POS) Tagging. POS tagging is the process of marking a word in the text to a particular part of speech based on its context and definition, and plays a crucial role in understanding how the word functions in the context, such as noun, punctuation, adverb, etc. Gimpel et al. (2010) provide a POS tagger for English data, and SpaCy also offers an excellent model to extract POS tags.

Dependency Parsing. Dependency parsing is to analyze the grammatical relationship among words of a sentence. There has been lots of research on this task. The SpaCy has a greedy transition-based dependency parser which can return various dependence parse tags.

Linearization. Linearization is a process to represent the structured data as a flat string so that the data-to-text task can be cast in the text-to-text framework (Kale & Rastogi, 2021). In practice, most works discard the distinction between rows (entities) and still represent data as a sequence of elements (Rebuffel et al., 2020).

ROUGE. Because it is difficult to gauge the correctness of the summary, evaluation metrics for summarization models use word overlap with the ground-truth summary in the form of ROUGE scores (Kryscinski et al., 2018; Lin, 2014).

3 Model Description

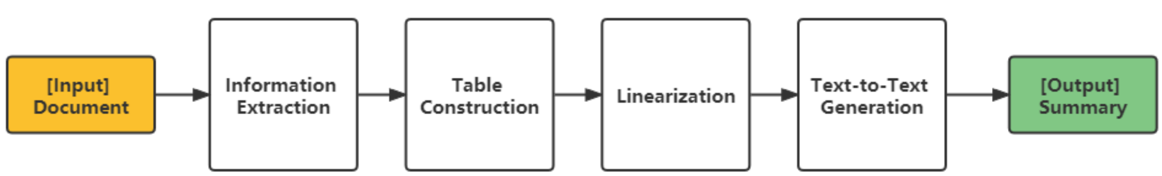

Figure 1: Overview of the pipeline

3.1 Information Extraction

The entities are firstly identified, and we rank these entities according to the number of times they are mentioned in the texts. The top three entities which are mentioned most frequently are selected. Then we apply coreference resolution to the raw text so that each coreferring mention in the raw text is replaced by the primary mention in the associated cluster.

Based on the top three entities, associated subjects and objects are found respectively using information like POS and DEP, and ranked according to the number of entities they are associated with. The most important words are those associated with all the top three entities, second with two entities, and the least with one.

Based on the top three entities and associated words, we score the sentences. The initial score for all sentences is zero. If a sentence includes one entity, add three points to its score. If it includes one most important word, add three points; two points for the second important and one for the least important. We use two dictionaries to store the information, one containing the content of sentences and the other scores of sentences. Finally, we sort sentences according to their scores and select the top 40% sentences.

We finally extract the sentence-level sentiment from the chosen sentences.

3.2 Table Construction

We store the extracted information in the form of a table. Each row represents one specific sentence and its corresponding crucial information. The table most likely contains the entities that are important in the original text and their relationships with each other.

For example, this is a raw text:

> A fire alarm went off at the Holiday Inn in Hope Street at about 04:20 BST on Saturday and guests were asked to leave the hotel. As they gathered outside they saw the two buses, parked side-by-side in the car park, engulfed by flames.
> One of the tour groups is from Germany, the other from China and Taiwan. It was their first night in Northern Ireland.
> The driver of one of the buses said many of the passengers had left personal belongings on board and these had been destroyed.
> Both groups have organised replacement coaches and will begin their tour of the north coast later than they had planned. Police have appealed for information about the attack.
> Insp David Gibson said: "It appears as though the fire started under one of the buses before spreading to the second. "While the exact cause is still under investigation, it is thought that the fire was started deliberately."

The constructed table is:

Table 1: The extracted information of sample text

Entity(Subject)	Verb	Object	Sentence-level sentiment
groups	said	tour	positive
guests	saw	buses	negative
One	is		positive

In this way, the representation of information is structured.

3.3 Linearization

Linearization transforms the table into a string, so that table-to-text generation can be transformed into a more simple text-to-text task. After linearization, each raw of the table is reorganized as <subject>xxx<verb>xxx<direct object>xxx<sentence-level sentiment>xxx where xxx stands for the data.

Therefore, for the first entity in figure 3, the string after linearization is:

> <subject>groups<verb>said<object>tour<sentence-level sentiment>positive<subject>guests<verb>saw <object>buses<sentence-level sentiment>negative<subject>One<verb>is<object><sentence-level sentiment>positive

3.4 Text-to-Text Generation

We adopted and fine-tuned the T5-small model mentioned in the related work. It is a text-to-text transformer with transfer learning and can be used for abstractive text summarization.

4 Experiments: Abstractive Text Summarization

4.1 Dataset

Xsum (Narayan et al., 2018) is an extreme summarization dataset consisting of BBC articles and accompanying single-sentence summaries. The output is a short, one-sentence news summary answering the question "What is the article about?". This dataset contains 203,045 documents for training, 11,332 for validation and 11,334 for testing.

4.2 Training Details

Firstly, We do some preprocessing on the documents in Xsum dataset. For each document, we extract the critical information, construct the table, and do linearization as mentioned in Model description to get a string from each document, which contains interpretable crucial information of a document. Then we stitch this string in front of the original text. After mapping such operation to all the documents, we get a preprocessed Xsum dataset.

Secondly, We train and validate T5-small with a batch size of 4. We set a constant learning rate of 2e-5, a constant weight decay of 0.01, and a constant dropout rate of 0.1. The maximum input length is limited to 512 tokens, and the output summary is limited to 128 tokens. All the model parameters are updated in the process.

4.3 Quantitative Evaluation

In the evaluation, we rely on metrics used by prior work. We evaluated summarization quality automatically using ROUGE (Lin, 2004). We report unigram and bigram overlap (ROUGE-1 and ROUGE-2) as a means of assessing informativeness, the longest common subsequence (ROUGE-L and ROUGE-Lsum) as a means of assessing fluency (Liu & Lapata, 2019). Below is the comparison of quantitative evaluation result between the T5-small fine-tuned in our way and some other models.

Table 2: Performance comparison of T5-small fine-tuned in our way to some other models on the same dataset

Model	Rouge-1	Rouge-2	Rouge-L
Baseline:Lead-3	16.30	1.6	11.95
PtGen-Covg	28.10	8.02	21.72
Seq2Seq	28.42	8.77	22.48
T5-small fine-tuned in our way	**28.69**	**8.17**	**22.83**

We can see that compared to the baseline and some other models, the T5-small fine-tuned in our way has an excellent performance in text summarization tasks.

For the text we showed in our model description, we got the summary:

> The driver of one of the buses said many of the passengers had left personal belongings on board and these were destroyed.

Therefore, our model did a good job in abstractive text summarization.

5 Conclusion and Future Work

This study proposed a novel approach for the abstractive text summarization task using structured input representation and fine-tuned T5 model. We found that this method leads to great results, allowing for better interpretability and control in the generation process by inspecting the structured input representation.

We noticed that pre-trained models like T5 could not guarantee constraint satisfaction. However, the newly proposed Mention Flags models can trace whether lexical constraints are satisfied in the generated outputs and generate tokens until all conditions are satisfied (Wang et al., 2021). Since our model allows for better interpretability and control in the generation process of summarization models, the MF models can be combined to help perform constrained generation to force the appearance of relevant entities and associated words into the output. We hope to extend this work in the future.

Moreover, though we used tables to represent the structured information in our work, we still appreciate the extractive graph models like EdgeSumm and Opinosis. As the Graph Transformer generalizes Transformer for arbitrary graphs, it is possible to apply it to the graphs extracted by EdgeSumm or Opinosis to get abstractive summaries. We hope to accomplish this work in the future and compare its results with our current work.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. (2017). Attention Is All You Need. *Advances in neural information processing systems*, pp.5998-6008.

[2] Chin-Yew Lin. (2014). Rouge: A package for automatic evaluation of summaries. *Text summarization branches out*, pp.74-81.

[3] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, Patrick Gallinari. (2020). A Hierarchical Model for Data-to-Text Generation. *Advances in Information Retrieval, 12035*, 65.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*

[5] Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, Yue Zhang. (2020). What Have We Achieved on Text Summarization? *arXiv preprint arXiv:2010.04529.*

[6] Frermann, Lea, Alexandre Klementiev.(2019). Inducing document structure for aspect-based summarization. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.6263-6273.

[7] Ganesan, Kavita; Zhai, ChengXiang; Han, Jiawei.(2010). Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. *Conference Proceedings and Workshop Materials - Computer Science*

[8] Isabel Groves.(2021). Automatically generating text from structured data. *CONVERSATIONAL AI / NATURAL-LANGUAGE PROCESSING*

[9] Jadhav, Aishwarya, and Vaibhav Rajan. (2018). Extractive Summarization with SWAP-NET: Sentences and Words from Alternating Pointer Networks. *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp.142-151.

[10] Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter Liu. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *International Conference on Machine Learning*, pp.11328-11339.

[11] Kale, Mihir, Abhinav Rastogi. (2020). Text-to-Text Pre-Training for Data-to-Text Tasks. *arXiv preprint arXiv:2005.10433*

[12] Laith Abualigah, Mohammad Qassem Bashabsheh, Hamzeh Alabool, Mohammad Shehab. (2020). Text Summarization: A Brief Review. *Recent Advances in NLP: The Case of Arabic Language*, pp.1-15.

[13] Liu, Yang, Mirella Lapata. (2019). Text Summarization with Pretrained Encoders. *arXiv preprint arXiv:1908.08345*

[14] Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, Hiroya Takamura. (2021). Towards Table-to-Text Generation with Numerical Reasoning. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp.1451-1465.

[15] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, Amr Ahmed. (2020). Big Bird: Transformers for Longer Sequences. *NeurlPS*

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461*

[17] Narayan, Shashi, Shay B. Cohen, Mirella Lapata. (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *arXiv preprint arXiv:1808.08745*

[18] Rohde, Tobias, Xiaoxia Wu, and Yinhan Liu. (2021). Hierarchical Learning for Generation with Long Source Sequences. *arXiv preprint arXiv:2104.07545*

[19] Syed, Ayesha Ayub, Ford Lumban Gaol, and Tokuro Matsuo. (2021). A Survey of the State-of-the-Art Models in Neural Abstractive Text Summarization. *IEEE Access*, 9, 13248-13265.

[20] Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, Yulia Tsvetkov. (2020). StructSum: Summarization via Structured Representations. *arXiv preprint arXiv:2003.00576*

[21] Vijay Prakash Dwivedi, Xavier Bresson. (2012). A Generalization of Transformer Networks to Graphs. *arXiv preprint arXiv:2012.09699*

[22] Wafaa S.El-Kassas, Cherif R.Salama, Ahmed A.Rafea, Hoda K.Mohamed.(2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing & Management*, 57(6): 102264.

[23] Wojciech Kryściński, Romain Paulus, Caiming Xiong, Richard Socher. (2018). Improving Abstraction in Text Summarization. *arXiv preprint arXiv:1808.07913*

[24] Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu. (2021). R-Drop: Regularized Dropout for Neural Networks. *arXiv preprint arXiv:2106.14448*

[25] Yang Liu, Mirella Lapata. (2019). Text summarizaiton with pretrained encoders. *arXiv preprint arXiv:1908.08345*

[26] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, Jie Tang. (2021). All NLP Tasks Are Generation Tasks: A General Pretraining Framework. *arXiv preprint arXiv:2103.10360*

[27] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, Xiang Zhou. (2020). Semantics-aware BERT for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 9628-9635.