A Study of Text Summarization with Graph Attention Networks

Ardestani, MohammadReza[1]

[1]Department of Computer Science
University of Lethbridge
ardestani@uleth.ca

Aug 19, 2024

Acknowledgment

Supervisor: I extend my deepest gratitude to Dr. **Yllias Chali**, whose continuous support and guidance have been the cornerstone of my journey.

Committee: My sincere appreciation goes to the committee members, Dr. **John Anvik** and Dr. **John Sheriff**, for their constructive feedback throughout several Progress Standing meetings.

Chair: I would like to express my gratitude to Dr. **Wendy Osborn**, for her unwavering support and her guidance on procedures.

University of Lethbridge, Alberta Innovates, NSERC: I'm immensely grateful for their generous support, allowing me to focus on my research.

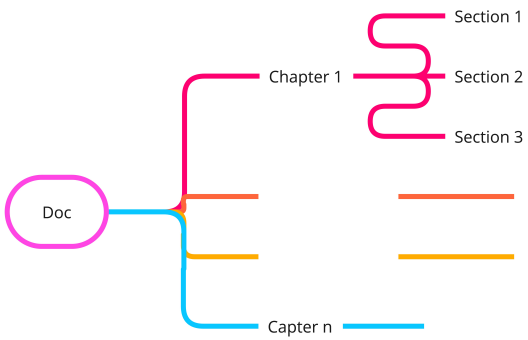
Compute Canada: Without the sponsorship of my supervisor for using the well-maintained High Performance Clusters of Compute Canada, conducting this research was not possible.

Internship Students: I had a pleasure to work with Taha Abbass and Riya Saxena whose contribution to the dataset processing were valuable.

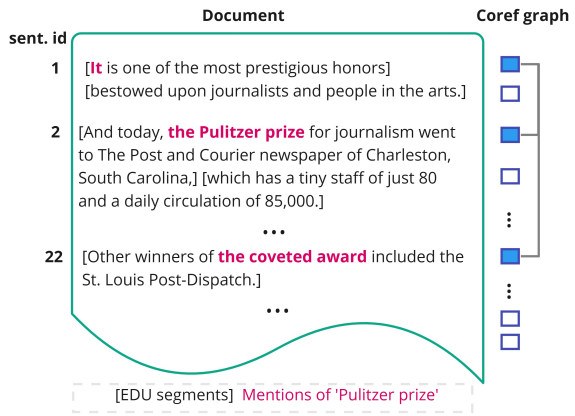
Table of Contents

Table of Contents

Overview

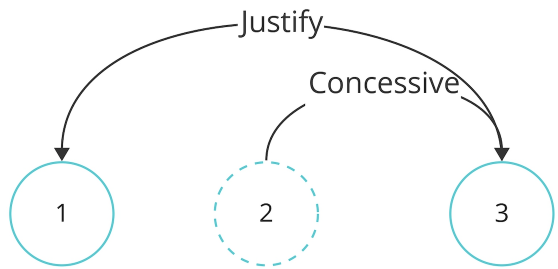


Chapter-Section graph



Coref graph

adopted from (Xu et al., 2020)



RST graph

adopted (Mann and Thompson, 1988)

- We naturally form a chapter-section graph of a document in our mind to understand that better.
- We used two similar graphs which facilitate summarization; RST (Mann and Thompson, 1988) & Coref (Sukthanker et al., 2018)

Table of Contents

Introduction

Summarization; an unsolved problem

Unlike Machine Translation (Popel et al., 2020), Automatic Text Summarization is remained far from achieving human-level performance (El-Kassas et al., 2021).

Transformer; a prevailing yet problematic architecture

Despite their success in machine translation, transformers struggle with more complex tasks like Dialogue Management and Text Summarization.

Graph; a guiding structure

Recent findings indicate that graphs can enhance the performance in T2T generation tasks (Liu and Wu, 2022).

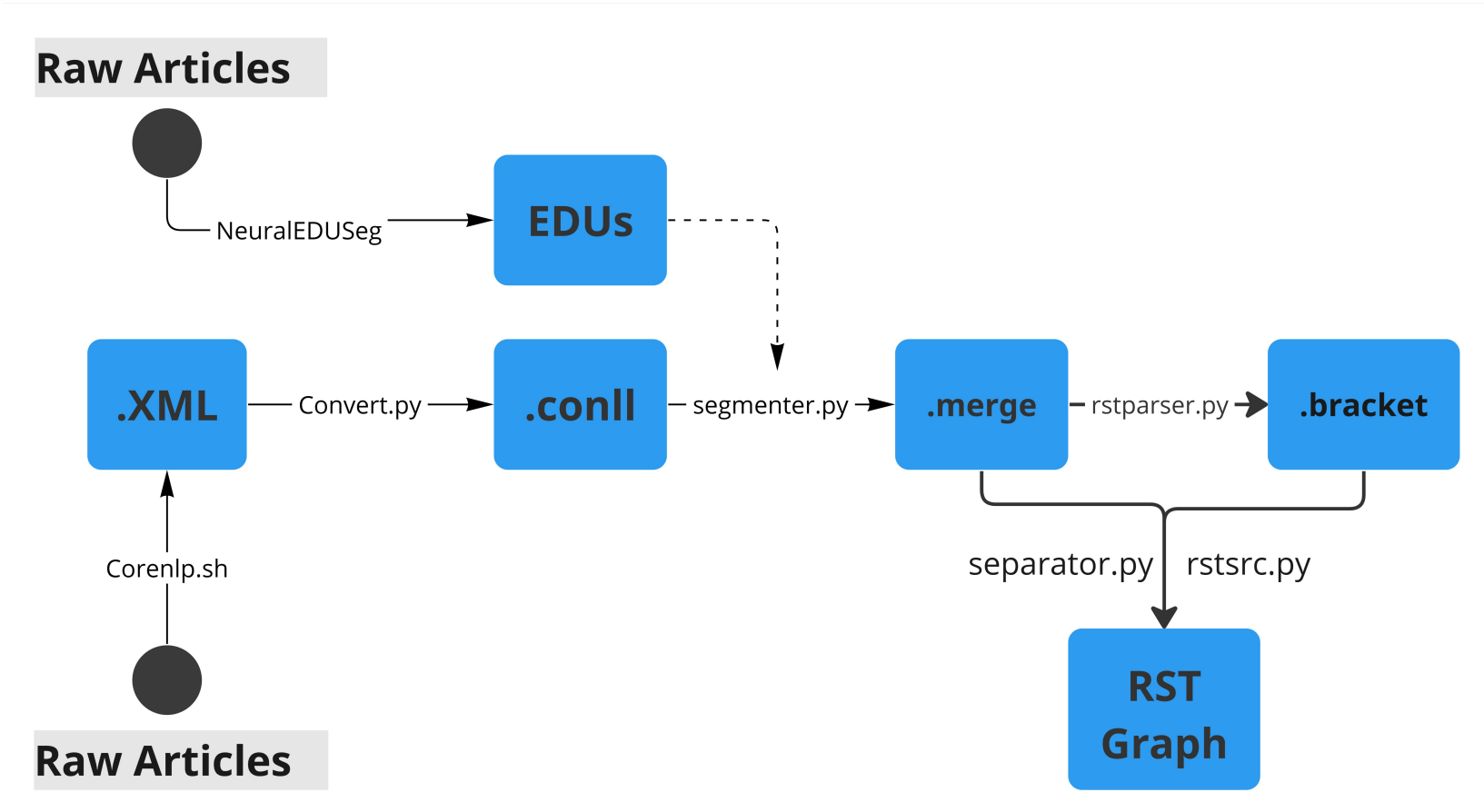
Table of Contents

Top Three Related Works

Model	Type	Distinctive Features
DiscoBERT (Xu et al., 2020)	Extractive	• Segments documents to smaller units (EDUs) • Leverages two graph structures for extraction
BART (Lewis et al., 2020)	Abstractive	• Trained using a denoising autoencoder approach • Combines the power of BERT and GPT • Handles a variety of text generation and comprehension tasks
BERTEXTABS (Liu and Lapata, 2019)	Hybrid	• Fine-tuned BERT for selecting important parts, then a trained transformer to generate the final summaries

Table of Contents

Dataset Prepration Pipeline



- We batched XSum Narayan et al. (2018) documents into 10K-size files and then distributed the process on eight servers to reduce the process time from 29 days to roughly 3 days.

Stats of the Generated Dataset

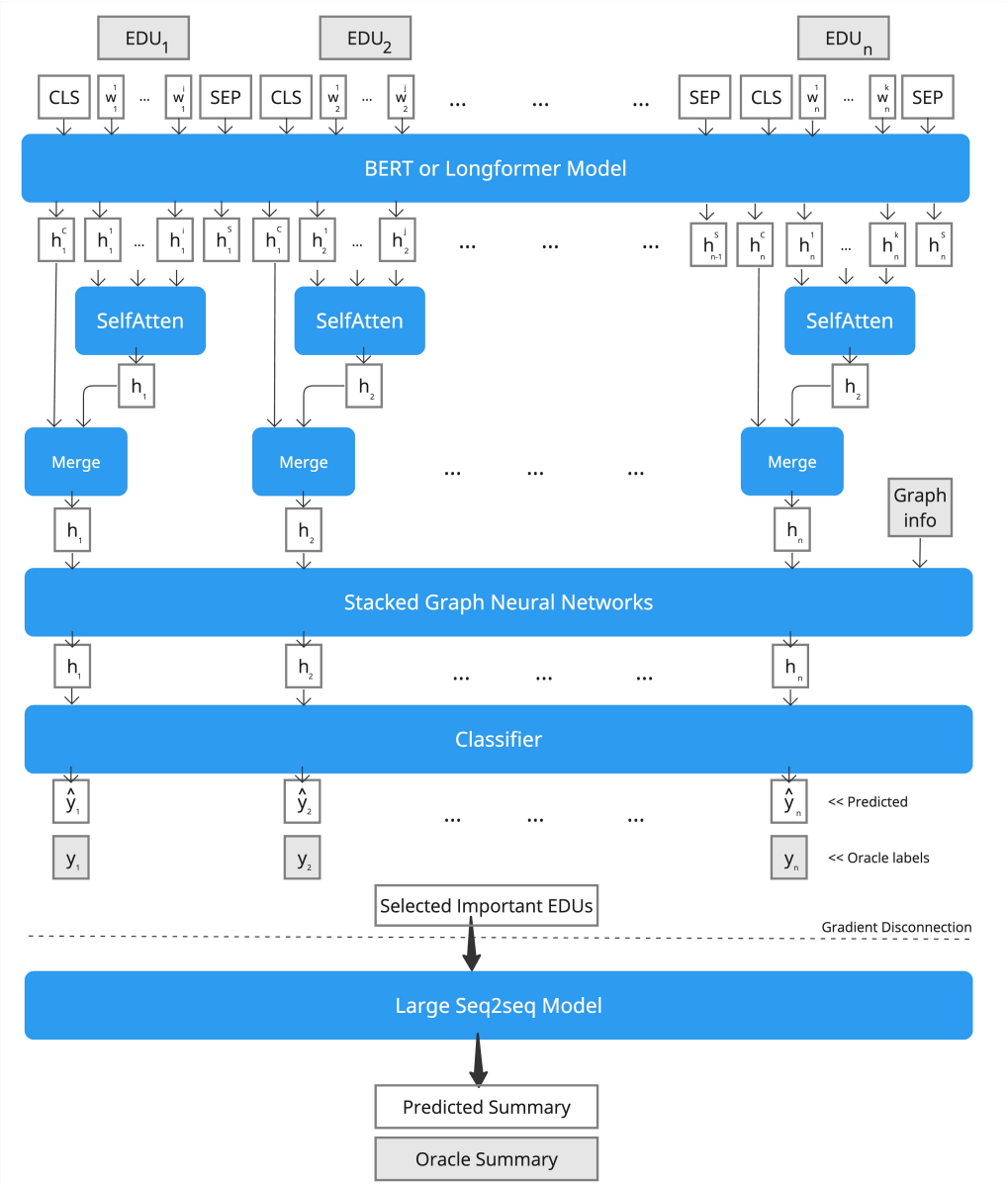
Table 3.8: Statistics of the processed XSum.

Dataset	Average		Empty			Total Records
	EDUs	Summary	Graph	Label	Both	
Train	52.35	21.10	121	1944	102	204017
Test	41.85	21.10	7	85	5	11333
Valid	41.39	21.13	10	96	8	11327

- Unlike CNN/DM, XSum summaries are short, averaging 21 words.
- Unlike CNN/DM, the summaries are extremely paraphrased.
- We undertook a costly dataset generation for two main reasons:
  - Bias reduction
  - Robustness

Table of Contents

Proposed Model Architecture

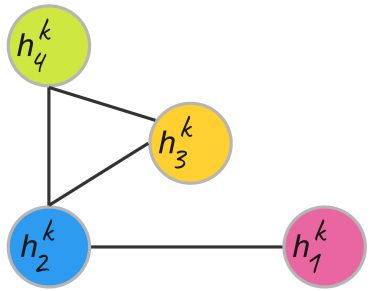


Our design & fine-tuning highlights:

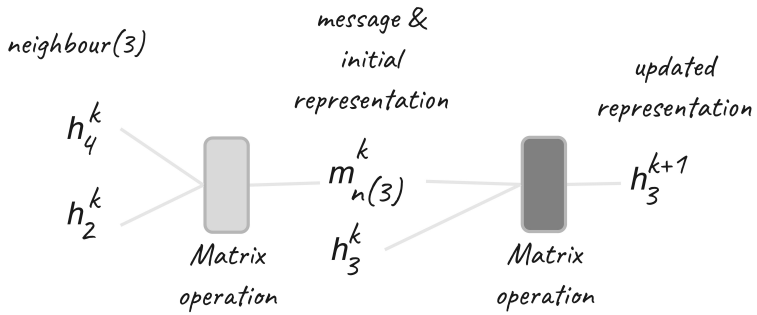
- Cached tokenized inputs, a repetitive process for all epochs in all experiments, resulted in reduced RAM usage & runtime
- Vectorized operations for faster matrix multiplication
- Utilized penalty to handle class imbalance
- Leveraged various ML methods to improve the selection F1 score
- Parallelized the fine-tuning to reduce runtime by 75%

These allowed us to efficiently conduct over 36 main experiments and 500 hyper-parameter searches.

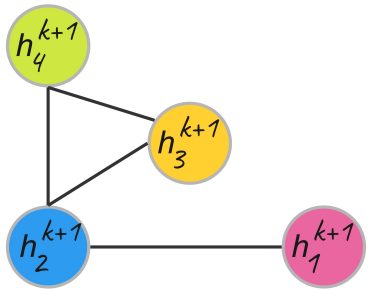
GNN



Graph at time k



Updating each node via message passing
(3rd node selected for showing the operations)



Graph at time k+1 with
updated node representation

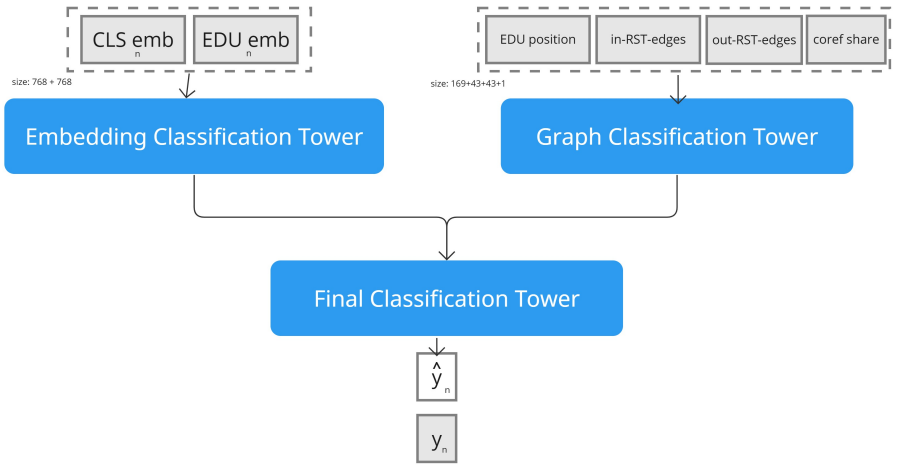
Graph learning includes two main operations (Hamilton, 2020):

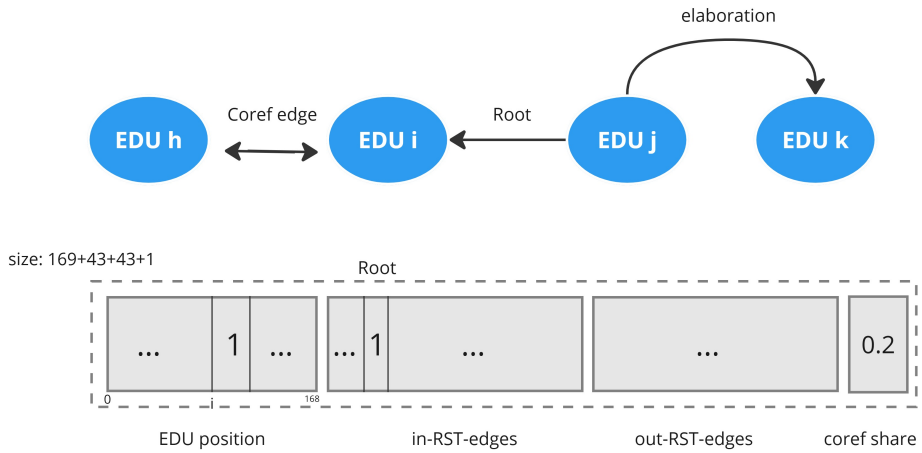
$$m_{N(u)}^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_v^{(k)}, \forall v \in N(u)\}\right) \tag{1}$$

$$h_u^{(k+1)} = \text{UPDATE}^{(k)}\left(h_u^{(k)}, m_{N(u)}^{(k)}\right) \tag{2}$$

Besides GAT, we also experimented with different GNN architectures, namely, GCN.

MLP Variation & Input





- As numerically shown in the next four table of results, our MLP performances mainly better than GAT for incorporating graph information
- This superiority suggest that not always a complicated architecture outperform its simple alternatives

Table of Contents

Our 10 Encoder Models on CNNDM

Table 5.1: Test results of the encoder models on CNNDM Dataset.

Model	Precision	Recall	F1 Score
Longformer without GAT	28.98	56.94	38.41
Longformer with GAT for RST	28.75	54.58	37.66
Longformer with GAT for Coref	26.57	57.26	36.30
Longformer with GAT for RST and Coref	27.62	57.94	37.40
Longformer with MLP for RST and Coref	29.81	54.94	38.65
BERT without GAT	28.61	65.00	39.74
BERT with GAT for RST	29.63	61.64	40.02
BERT with GAT for Coref	29.63	61.84	40.06
BERT with GAT for RST and Coref	29.74	61.89	40.18
BERT with MLP for RST and Coref	29.64	62.17	40.15

- Early versions of "Longformer w.o. GAT" model had $\approx$ 35% F1 score. We used various ML methods to increase it to 38.41%
- Longformer-based models Beltagy et al. (2020) have triple the input length of BERT Devlin et al. (2019), impacting comparisons between these two models.

Our 6 Encoder Models on XSum

Table 5.2: Test results of the encoder models on XSum Dataset.

Model	Precision	Recall	F1 Score
Longformer without GAT	30.71	50.04	38.06
Longformer with GAT for RST	30.89	48.06	37.61
Longformer with MLP for RST	30.86	49.74	38.09
BERT without GAT	31.11	51.48	38.78
BERT with GAT for RST	31.29	50.69	38.70
BERT with MLP for RST	32.52	48.85	39.05

- MLP-based models show superior performance than their counterparts, showing that incorporating graph information using MLP can help the selection.

Our Decoder Models on CNNDM

Table 5.3: Test results of encoder-decoder models on CNNDM Dataset.

Model	R-1	R-2	R-L
Longformer without GAT	43.60 ± 0.23	20.77 ± 0.25	40.60 ± 0.22
Longformer with GAT for RST	43.01 ± 0.23	20.35 ± 0.25	40.00 ± 0.21
Longformer with GAT for Coref	43.04 ± 0.22	20.32 ± 0.25	39.98 ± 0.24
Longformer with GAT for RST and Coref	43.30 ± 0.23	20.55 ± 0.23	40.30 ± 0.23
Longformer with MLP for RST and Coref	43.73 ± 0.22	20.91 ± 0.26	40.72 ± 0.24
BERT without GAT	43.43 ± 0.24	20.62 ± 0.24	40.40 ± 0.23
BERT with GAT for RST	43.37 ± 0.22	20.65 ± 0.25	40.34 ± 0.23
BERT with GAT for Coref	43.40 ± 0.23	20.64 ± 0.24	40.39 ± 0.21
BERT with GAT for RST and Coref	43.34 ± 0.23	20.64 ± 0.25	40.34 ± 0.24
BERT with MLP for RST and Coref	43.42 ± 0.23	20.70 ± 0.25	40.34 ± 0.23

Average ROUGE scores are obtained by 1000 replicates and ± is followed by an estimated margin of error under 95% confidence (Koehn, 2004).

- Amongst 10 different models, "Longformer with MLP" is performing better, indicating the positive effect of using graph using our MLP approach for long documents.

Our Decoder Models on XSum

Table 5.4: Test results of encoder-decoder models on XSum Dataset.

Model	R-1	R-2	R-L
Longformer without GAT	36.56 ± 0.25	14.28 ± 0.23	28.29 ± 0.23
Longformer with GAT for RST	36.24 ± 0.26	14.14 ± 0.23	28.14 ± 0.24
Longformer with MLP for RST	36.82 ± 0.27	14.66 ± 0.24	28.67 ± 0.26
BERT without GAT	36.41 ± 0.28	14.23 ± 0.25	28.37 ± 0.27
BERT with GAT for RST	35.33 ± 0.25	13.30 ± 0.23	27.45 ± 0.25
BERT with MLP for RST	35.97 ± 0.25	13.95 ± 0.24	27.90 ± 0.26

Average ROUGE scores are obtained by 1000 replicates and ± is followed by an estimated margin of error under 95% confidence.

- Note that we are only using RST graph for XSum, due to time constrants.

Outperforming Industry Leaders on CNNDM

Table 5.5: Comparing our best model with previous SOTA models on CNNDM.

Model	R-1	R-2	R-L
Oracle			
Lead3	40.42	17.62	36.67
Oracle (EDUs)	61.61	37.82	59.27
Extractive			
DiscoBERT (Xu et al., 2020)	43.38	20.44	40.21
DiscoBERT w Coref	43.58	20.64	40.42
DiscoBERT w RST	43.68	20.71	40.54
DiscoBERT w RST & Coref	**43.77**	20.85	40.67
BERTSUMEXT (Liu and Lapata, 2019)	43.25	20.24	39.63
Abstractive			
BERTSUMABS (Liu and Lapata, 2019)	41.72	19.39	38.76
BART-base† (Lewis et al., 2020)	41.44	18.49	38.32
Hybrid			
BERTSUMEXTABS (Liu and Lapata, 2019)	42.13	19.60	39.18
Longformer with MLP for RST and Coref (ours)	43.73	**20.91**	**40.72**

Comparison of our Best Model on XSum with Others

Table 5.6: Comparing our best model with previous SOTA models on XSum.

Model	R-1	R-2	R-L
Oracle			
LEAD1 †	16.30	1.60	11.95
Oracle (Best Sentence) (Liu and Lapata, 2019)	29.79	8.81	22.66
Oracle (Best EDUs) (ours)	36.03	11.47	30.86
Abstractive			
BERTSUMABS (Liu and Lapata, 2019)	38.76	16.33	31.15
BART-large ‡ (Lewis et al., 2020)	**42.34**	**18.45**	**32.40**
Zeroshot BART-large on oracle EDUs	31.94	11.10	24.62
Finetuned BART-large on oracle EDUs	42.34	18.44	32.40
Hybrid			
BERTSUMEXTABS (Liu and Lapata, 2019)	38.81	16.50	31.27
Longformer with MLP for RST (ours)	36.82	14.66	28.67

† The term "Lead1" refers to the first sentences of the input documents.
‡ We generated this result based on their provided fine-tuned model.

- For short documents, using only a decoder might work better.
- We used a dataset with the most contrastive characteristics to reveal the limitations of our approach.

Effectiveness of Extraction on CNNDM vs XSum

Table 5.7: Comparing the performance of extractive approaches on CNNDM and XSum.

Model	R-1	R-2	R-L
CNNDM			
BART-base	41.44	18.49	38.32
Oracle (EDUs)	**61.61** (↑)	**37.82** (↑)	**59.27** (↑)
Zeroshot BART-base on oracle EDUs	53.43 (↑)	32.20 (↑)	51.05 (↑)
Finetuned BART-base on oracle EDUs	60.89 (↑)	37.59 (↑)	57.47 (↑)
XSum			
BART-large	**42.34**	**18.45**	**32.40**
Oracle (EDUs) (ours)	36.03 (↓)	11.47 (↓)	30.86 (↓)
Zeroshot BART-large on oracle EDUs	31.94 (↓)	11.10 (↓)	24.62 (↓)
Finetuned BART-large on oracle EDUs	42.34 (↓)	18.44 (↓)	32.40 (↓)

- Unlike CNNDM dataset, extracting important parts of the XSum dataset does not benefit as much from simply using the entire document as input for the decoder.

Table of Contents

Conclusion & Future Directions

Outperformed previous SOTA models with simpler graph architecture

Used a simple MLP rather than a complicated architecture, which outperformed previous SOTA on CNNDM dataset.

Established a benchmark for graph-based summarization

Selected a distinctive dataset, annotated it with RST graphs, and established it as a benchmark for future researches.

Identified promising future directions

Looking forward, We recommend using alternative metrics to ROUGE or addressing gradient disconnection by integrating documents and graph information into an end-to-end model.

Q&A Section

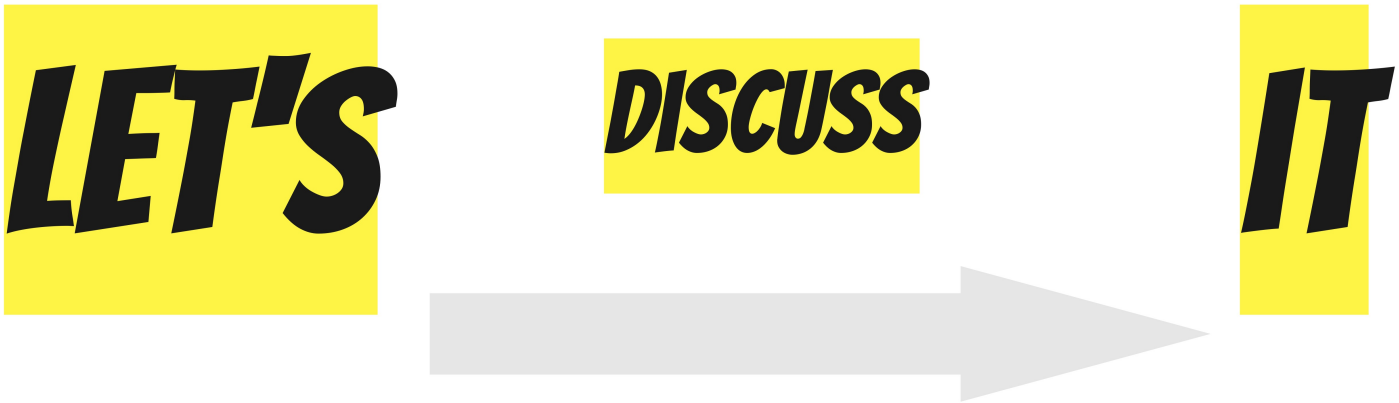


Table of Contents

References I

I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.113679. URL https://www.sciencedirect.com/science/article/pii/S0957417420305030.

W. L. g. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3): 1–159, 2020.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

B. Liu and L. Wu. Graph neural networks in natural language processing. In L. Wu, P. Cui, J. Pei, and L. Zhao, editors, *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 463–481. Springer Singapore, Singapore, 2022.

Y. Liu and M. Lapata. Text summarization with pretrained encoders. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1387. URL https://aclanthology.org/D19-1387.

W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988. doi: doi:10.1515/text.1.1988.8.3.243. URL https://doi.org/10.1515/text.1.1988.8.3.243.

S. Narayan, S. B. Cohen, and M. Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

References II

M. Popel, M. Tomkova, J. Tomek, Ł. Kaiser, J. Uszkoreit, O. Bojar, and Z. Žabokrtský. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1): 4381, Sep 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18073-9. URL https://doi.org/10.1038/s41467-020-18073-9.

R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu. Anaphora and coreference resolution: A review, 2018.

J. Xu, Z. Gan, Y. Cheng, and J. Liu. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.