Compressed Heterogeneous Graph for Abstractive Multi-document Summarization

Miao Li, Jianzhong Qi, and Jey Han Lau

School of Computing and Information Systems,
The University of Melbourne, Australia

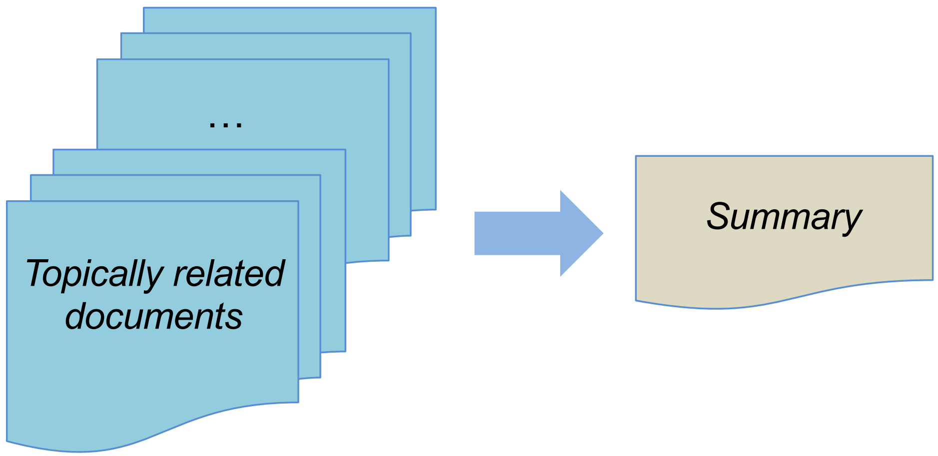
The 37th AAAI Conference on Artificial Intelligence, February 7-14, 2023

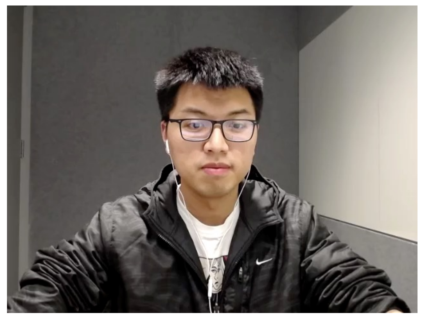
Task definition and background

- Abstractive multi-document summarization (MDS)
  - Input: topically related documents
  - Output: a summary
- Summarizing multiple documents in an abstractive fashion

$$\mathrm{p}(\hat{z}|\mathcal{D}) = \prod_{i=0}^{T} \mathrm{p}(\hat{w_i}|\mathcal{D}, \hat{w_0}, \hat{w_1}, \ldots, \hat{w_{i-1}})$$

- A wide range of applications, e.g.,
  - Creating news digests (Fabbri et al. 2019)
  - Summarizing product reviews (Gerani et al. 2014)





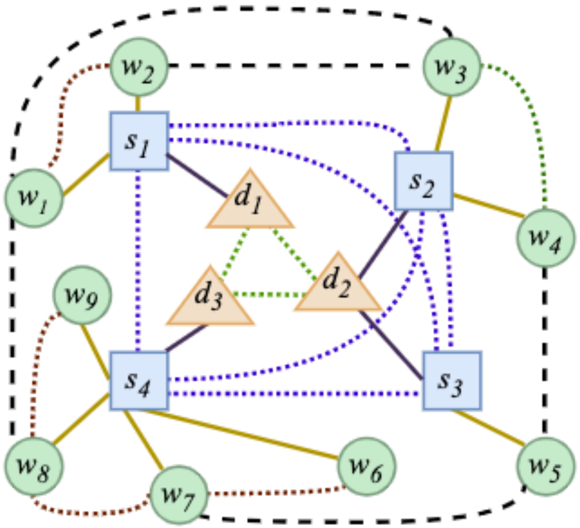
Related work and challenges

- PLM-based MDS
  - General-purpose PLMs, e.g., BART, Longformer, and T5
  - Tailored-purpose PLMs, e.g.,
    - PEGASUS (Zhang et al. 2020a)
    - PRIMERA (Xiao et al. 2022)
- Drawbacks
  - Limited to learn **cross-document relationships** because of the flat concatenation of source documents

- Graph-based MDS
  - Only a handful of this type of models for abstractive MDS
  - Graphs of paragraphs (Li et al. 2020)
  - Hierarchical graphs based on the document structure (Jin et al. 2020)
- Drawbacks
  - Only leverage **homogeneous** graphs
  - without considering different edge types of graphs
  - while the cluster of documents should be **heterogeneous**

Our solution of HGSum

- ❑ Using a heterogeneous graph to represent each cluster of source documents



w: word; *s*: sentence; *d*: document

- ❑ Borrowing ideas from hierarchical graph pooling
  - ○ Node dropping to generate a small-sized graph

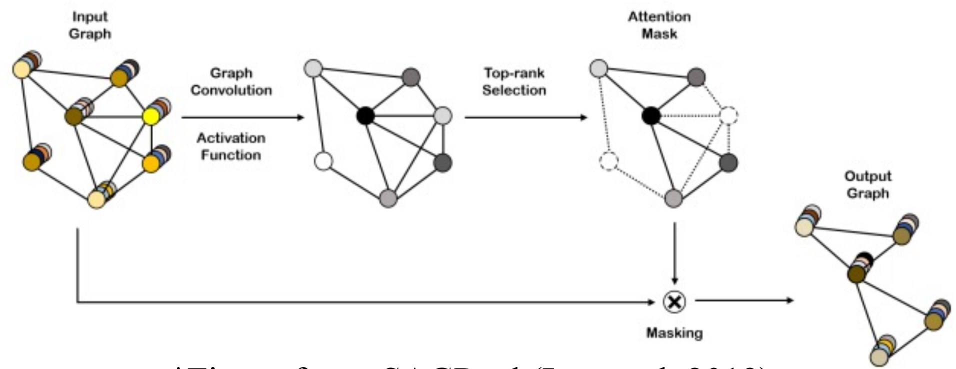


*Figure from SAGPool (Lee et al. 2019)

Incorporating heterogeneous GNN into seq2seq

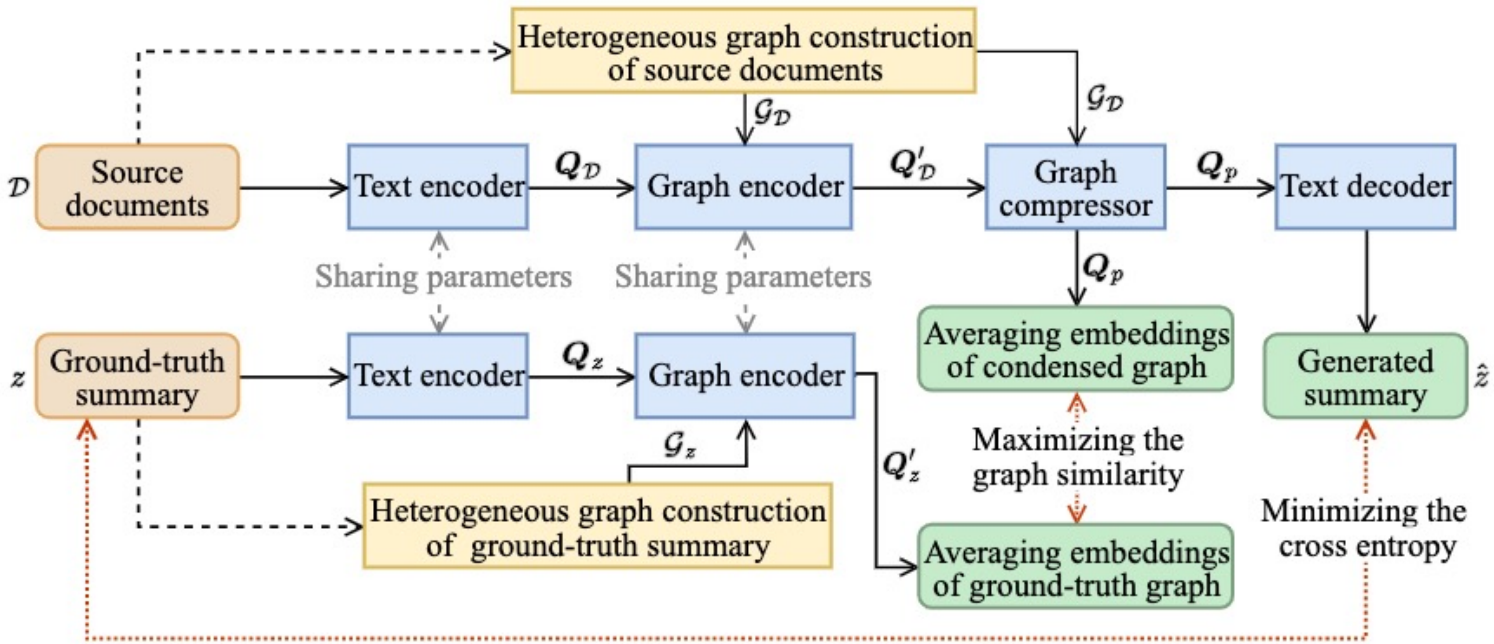
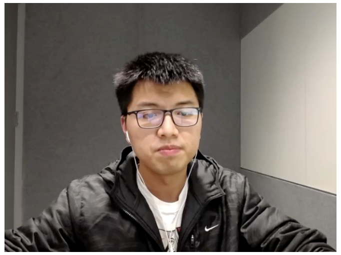


Figure 2: The HGSUM architecture: There are four main components: (1) text encoder (initialised using PRIMERA weights); (2) graph encoder; (3) graph compressor; and (4) text decoder (initialised using PRIMERA weights).



Incorporating heterogeneous GNN into seq2seq

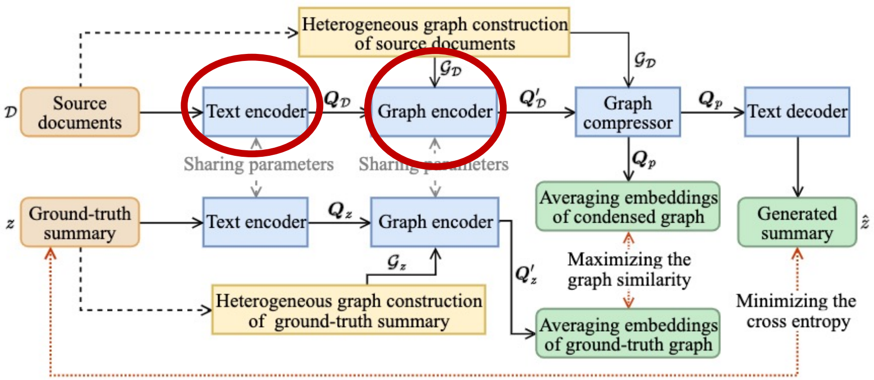


Figure 2: The HGSUM architecture: There are four main components: (1) text encoder (initialised using PRIMERA weights); (2) graph encoder; (3) graph compressor; and (4) text decoder (initialised using PRIMERA weights).

- Graph encoder, Multi-channel GAT

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{U}\boldsymbol{H}_i^{(l)} \qquad \boldsymbol{h}_i^{(l),c} = \Big\Vert_{m=1}^{M} \sigma\Big(\sum_{j\in\mathcal{N}_i^c} \alpha_{ij}^{m,c} \boldsymbol{W}^{m,c} \boldsymbol{h}_j^{(l),c}\Big)$$

$$\boldsymbol{H}_i^{(l)} = \Big\Vert_{c=1}^{C} \boldsymbol{h}_i^{(l),c}$$

$$\alpha_{ij}^{m,c} = \frac{\exp(d_{ij}^{m,c})}{\sum_{k\in\mathcal{N}_i^c} \exp(d_{ik}^{m,c})}$$

$$d_{ij}^{m,c} = \sigma\big(e_{ij} \cdot \boldsymbol{w}_{m,c}^{\top}[\boldsymbol{W}^{m,c}\boldsymbol{h}_i^{(l),c} \Vert \boldsymbol{W}^{m,c}\boldsymbol{h}_j^{(l),c}]\big)$$

- Text encoder

$$\boldsymbol{Q}_{\mathcal{D}} = \mathrm{longformer}(\mathcal{D})$$
$$\boldsymbol{Q}_z = \mathrm{longformer}(z)$$

Incorporating heterogeneous GNN into seq2seq

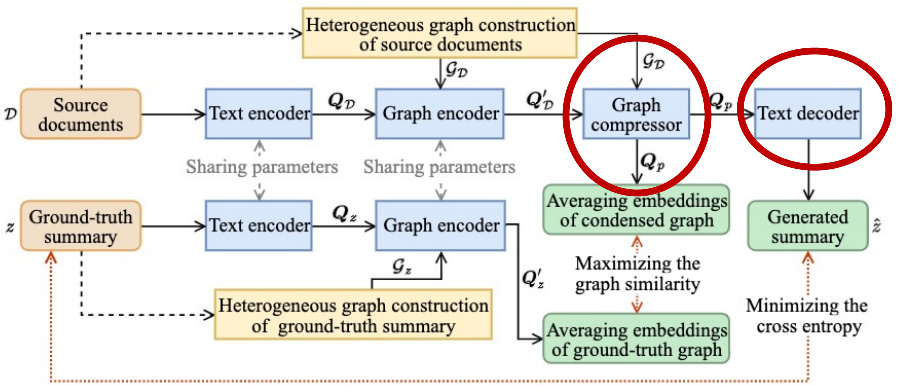


Figure 2: The HGSUM architecture: There are four main components: (1) text encoder (initialised using PRIMERA weights); (2) graph encoder; (3) graph compressor; and (4) text decoder (initialised using PRIMERA weights).

- Graph compressor

$$\boldsymbol{t} = \text{softmax}(\text{MGAT}(\boldsymbol{Q}_{\mathcal{D}}, \mathcal{G}_{\mathcal{D}}) \cdot \boldsymbol{r})$$

$$\mathcal{I}_s = \text{top-k}(\boldsymbol{t}, k, \mathcal{G}_{\mathcal{D}})$$

$$\mathcal{I} = \text{extend}(\mathcal{I}_s, \mathcal{G}_{\mathcal{D}})$$

- Text decoder

$$\hat{z} = \text{transformer}(\boldsymbol{Q}_p)$$

Incorporating heterogeneous GNN into seq2seq

- Multi-task training

$$\mathcal{L} = \beta\mathcal{L}_{ce} + (1-\beta)\mathcal{L}_{gs}$$

$$\mathcal{L}_{ce} = -\frac{1}{T}\sum_{i=1}^{T} w_i \log \hat{w}_i$$

$$\mathcal{L}_{gs} = -\,\mathrm{sim}(\mathrm{avg}(\boldsymbol{Q}_p), \mathrm{avg}(\boldsymbol{Q}'_z))$$

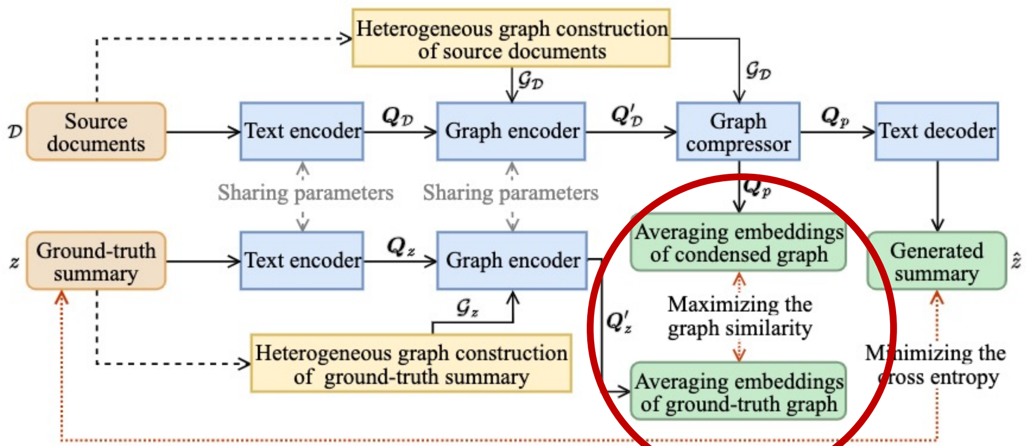


Figure 2: The HGSUM architecture: There are four main components: (1) text encoder (initialised using PRIMERA weights); (2) graph encoder; (3) graph compressor; and (4) text decoder (initialised using PRIMERA weights).

Experiments: main results

Model	MULTI-NEWS R-1	MULTI-NEWS R-2	MULTI-NEWS R-L	WCEP-100 R-1	WCEP-100 R-2	WCEP-100 R-L	ARXIV R-1	ARXIV R-2	ARXIV R-L
PEGASUS	47.70	18.36	43.62	42.43	17.33	32.35	44.21	16.95	38.87
LED	47.68	19.72	43.83	43.05	20.94	34.99	46.50	18.96	41.87
PRIMERA	49.40	20.51	45.35	43.11	**21.85**	35.89	47.24	20.24	42.61
MGSum	45.63	16.71	40.92	38.88	14.22	23.37	40.58	11.22	29.93
GraphSum	45.71	17.12	41.99	39.56	14.38	29.41	42.98	16.55	37.01
HGSUM (our model)	**50.64**	**21.69**	**45.90**	**44.21**	21.81	**36.21**	**49.32**	**21.30**	**44.50**
Performance gain	+2.51%	+5.75%	+1.21%	+2.55%	-0.18%	+0.89%	+4.40%	+5.24%	+4.44%

Table 3: Model performance on summarizing MULTI-NEWS, WCEP-100, and ARXIV in terms of F1 of ROUGE scores. The best performance results are in boldface, while the second best is underlined.

- ✓ HGSum outperforms most of benchmark systems
- ✓ PLM-based models seems consistently better than previous graph-based models

Experiments: ablation study

- ✓ Removing different components result in a performance drop over all metrics
- ✓ Dropping the multi-task objective leads to the largest degradation in model performance

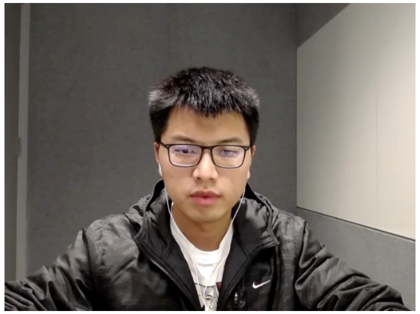
Model	R-1	R-2	R-L	BScore
HGSUM	50.64	21.69	45.90	87.38
w/o MGAT	48.87	20.32	43.21	87.08
w/o graph compressor	49.00	20.38	45.01	86.92
w/o multi-task training	48.10	20.30	44.24	86.85

Table 5: Results of ablation study on MULTI-NEWS.

- ✓ Our model can be initialized by any pre-trained Transformers

Initialized by	R-1	R-2	R-L	BScore
random weights	18.99	27.86	16.88	79.32
LED	48.36	19.99	44.25	86.73
PRIMERA	50.64	21.69	45.90	87.38

Table 6: Summarization results of HGSUM with different initialization on MULTI-NEWS.

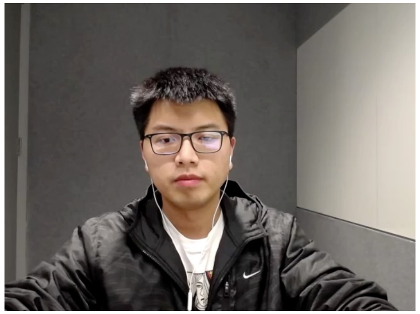


Takeaways

- Abstractive multi-document summarization
- Aim to incorporate cross-document relationships into seq2seq
  - Construct heterogeneous graphs to represent cross-document relationships
  - Propose the idea of compressed heterogeneous graphs to incorporate GNN into Transformer architecture
- Experiments show HGSum outperforms other strong baselines on three datasets

- Limitations
  - Use more better evaluation metrics like BERTScore and BARTScore to evaluate the model
  - Quantitatively evaluate whether the proposed model handles various cross-document relationships like contradicts

*For details, please refer to our paper



Questions & Answers

https://oaimli.github.io
miao4@student.unimelb.edu.au