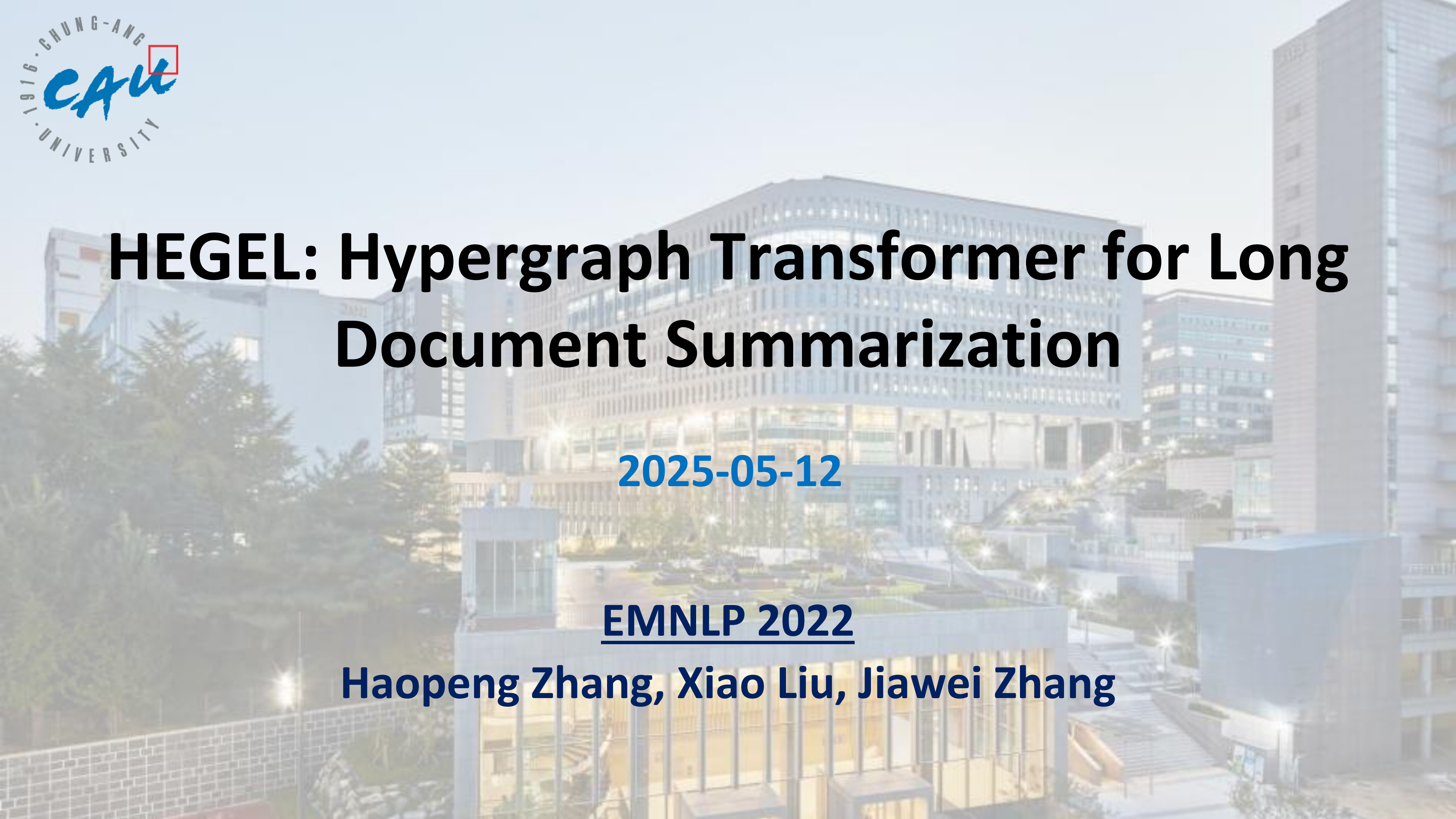


HEGEL: Hypergraph Transformer for Long Document Summarization

2025-05-12

EMNLP 2022

Haopeng Zhang, Xiao Liu, Jiawei Zhang

Content

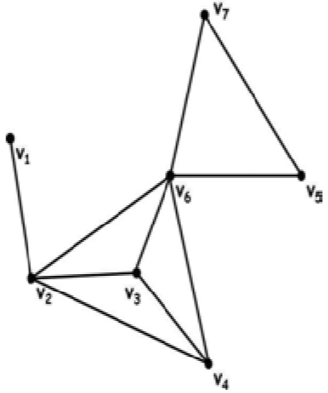
- ☐ Introduction
- ☐ Previous works
- ☐ Goal
- ☐ HEGEL
- ☐ Experiments
- ☐ Conclusion
- ☐ Appendix

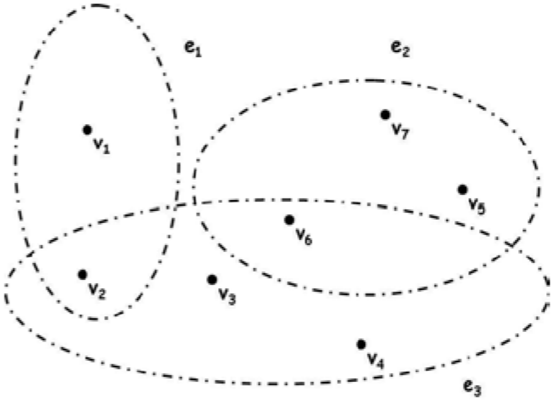
Introduction

☐ What is a Hypergraph?

- A graph in which an edge can connect more than two vertices
  - Edge is a subset of vertices

	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	1
v_3	0	0	1
v_4	0	0	1
v_5	0	1	0
v_6	0	1	1
v_7	0	1	0
v_2	1	0	1



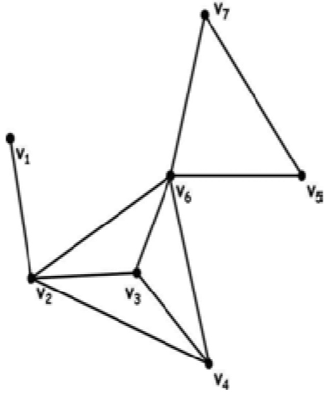


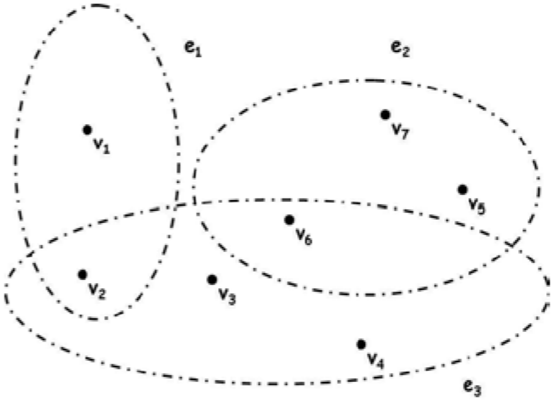
Introduction

☐ Why the hypergraph is important?

- The hypergraph can represent more complex relationships among the objects

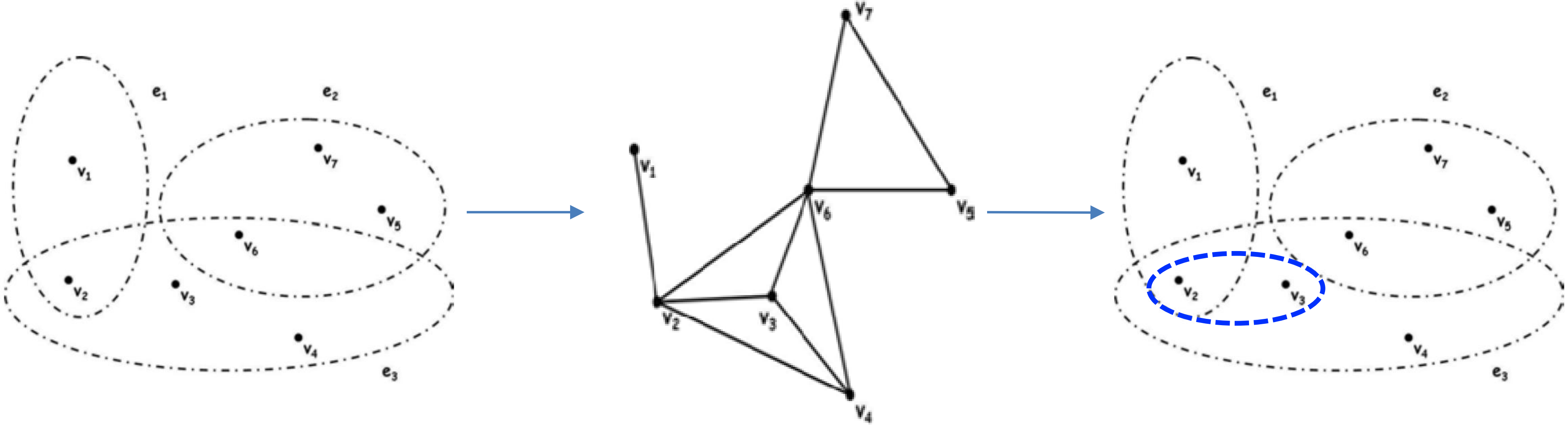
	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	1
v_3	0	0	1
v_4	0	0	1
v_5	0	1	0
v_6	0	1	1
v_7	0	1	0
v_2	1	0	1





Introduction

- **Why the hypergraph is important?**
  - Unable to recover hypergraphs from simple graphs
    → *Information loss!*

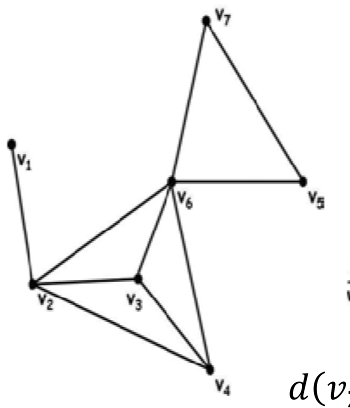


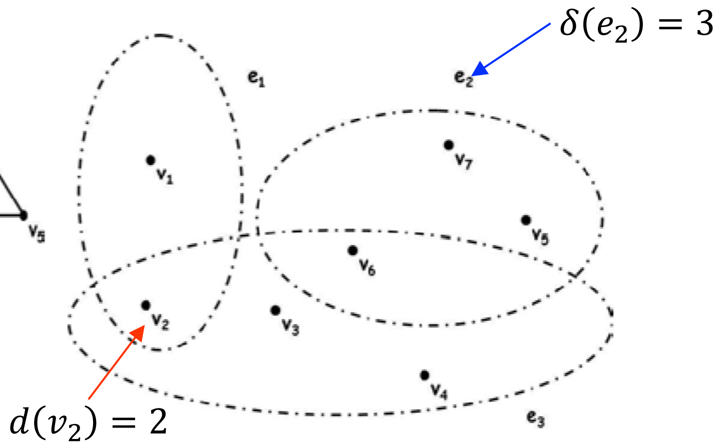
Introduction

☐ Preliminaries

- A hypergraph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{w})$
  - ☐ V : a finite set of objects, E : a family of subsets e of V s.t $\cup_{e \in E} = V$
  - ☐ w : a weight of hyperedge
  - ☐ $|V| \times |E|$ matrix H with entries $h(v, e) = 1$ if $v \in e$ and 0 otherwise
  - ☐ $d(v) = \sum_{e \in E} w(e) h(v, e)$, $\delta(e) = \sum_{v \in V} h(v, e)$

	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	1
v_3	0	0	1
v_4	0	0	1
v_5	0	1	0
v_6	0	1	1
v_7	0	1	0
v_2	1	0	1

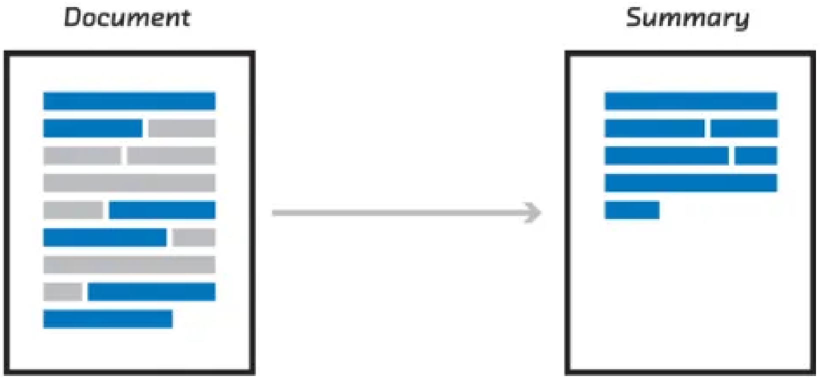




Introduction

☐ Extractive Summarization

- Generate a shorter version of a document
- Preserve the most salient information
- Directly extract relevant sentences from the original document



Extractive summarizers

Introduction

☐ Long Document ($e.g.$, Scientific paper) Summarization

- Long structured input
- Cover diverse topics
- Have richer structural information

→ Different for sequential models to capture

→ GNN-based approaches to model cross-sentence relations

Previous works

- **GNN-based approaches to model cross-sentence relations**
  - Represent a document with a sentence-level graph
  - Extractive Summarization → Node classification problem

Previous works

- GNN-based approaches to model cross-sentence relations
  - Inter-sentence cosine similarity graph
    - Configure edges based on cosine similarity between sentences with sentences
    - Calculate Sentence Salience

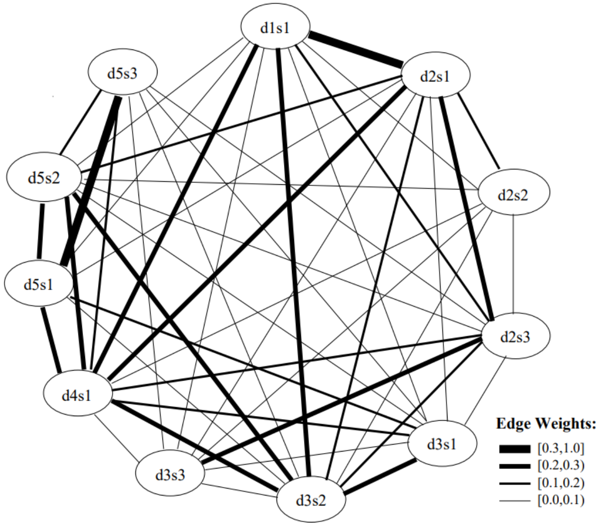
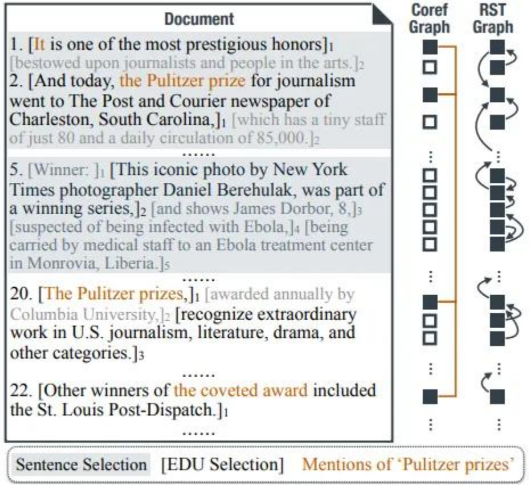


Figure 2: Weighted cosine similarity graph for the cluster in Figure 1.

Previous works

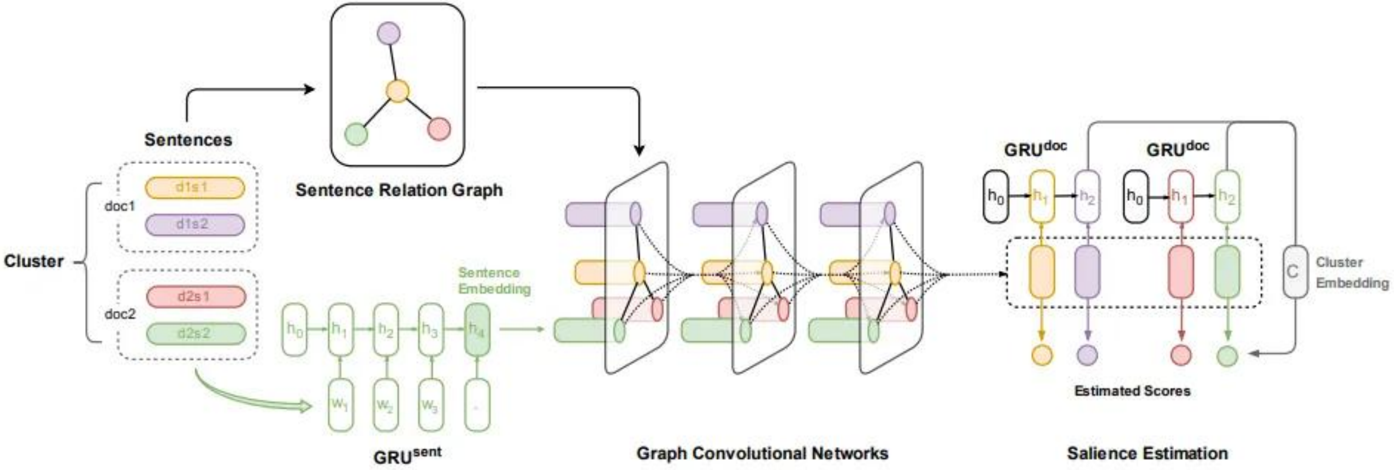
- ☐ **Rhetorical Structure Theory (RST) tree relation graph**
  - Segment the whole document into Element Discourse Units (EDUs)
    - ☐ Contiguous, adjacent and non-overlapping text spans
    - ☐ Nucleus (generally more central)
    - ☐ Satellite (less important in terms of content and grammatical reliance)
  - Provide local paragraph-level and document-level connections



Previous works

☐ Approximate Discourse Graph

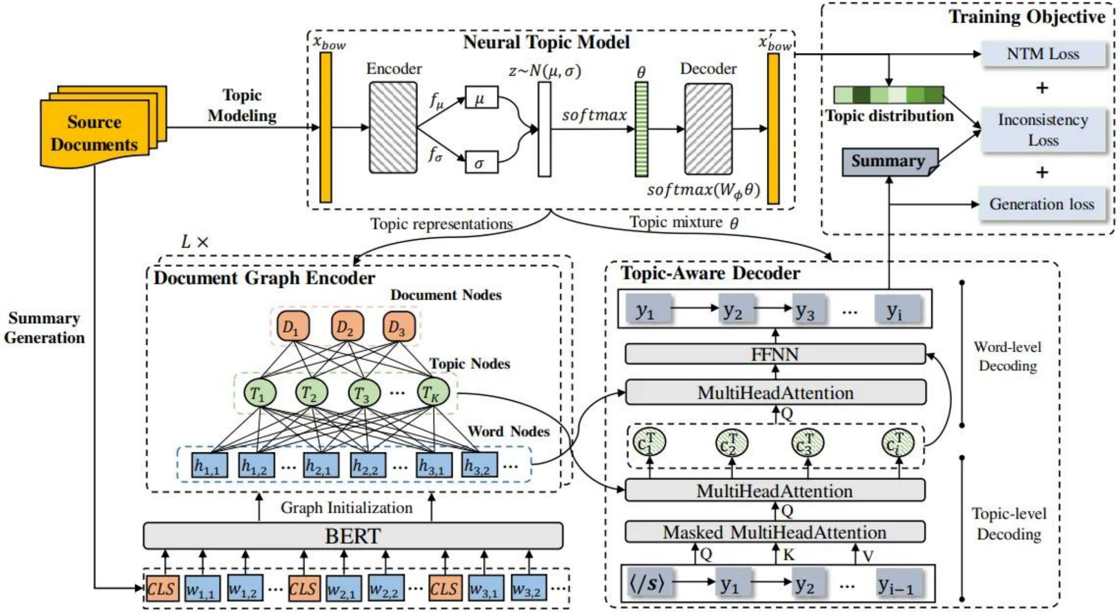
- Create Sentence Relation Graph based on ADG and Personalized Discourse Graph (PDG)



Previous works

☐ Topic-Sentence Graph

- Create a graph composed of Word, Topic, Document nodes

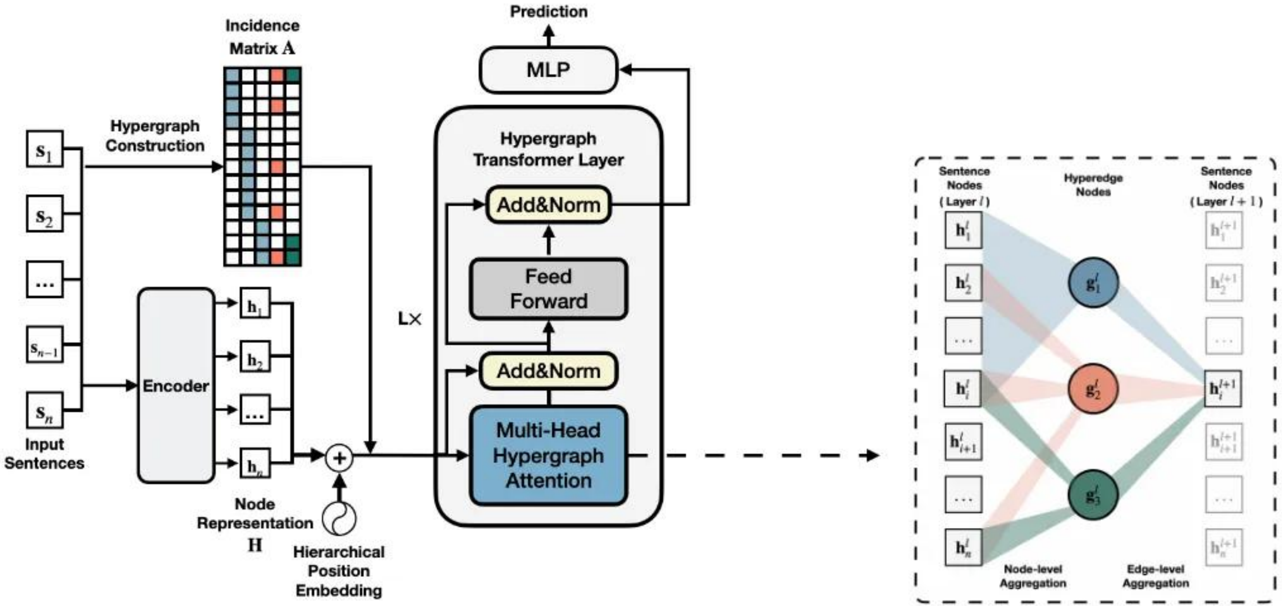


Previous works

- **Limitation of Existing GNN-based methods**
  - Only model the pairwise interaction between sentences
  - Incapable of fusing sentence interactions from different perspectives
    - Embedding similarity
    - Keywords coreference
    - Topical modeling from the semantic perspective
    - Section of rhetorical structure from the discourse perspecitve

Goal

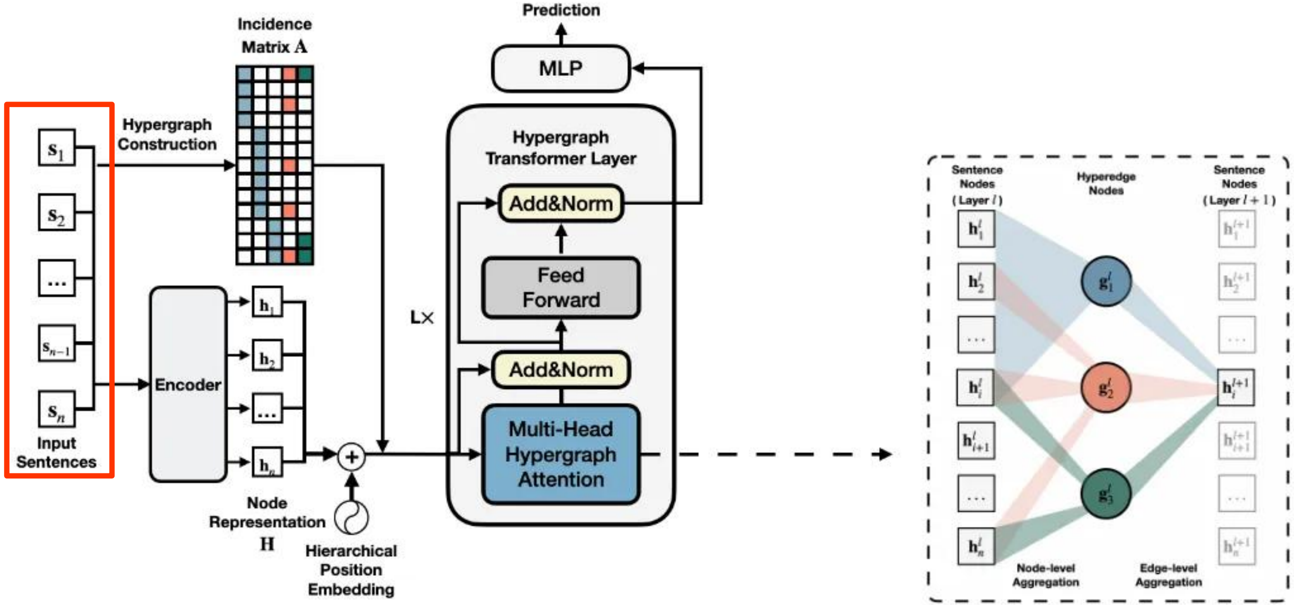
- ☐ **Model high-order cross-sentence relations with hypergraphs for extractive document summarization**



HEGEL

☐ Document as a Hypergraph

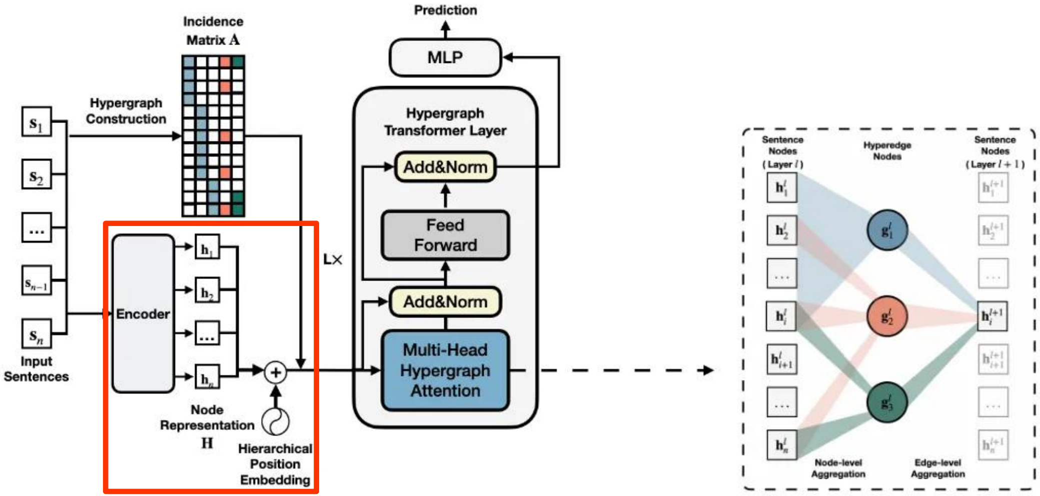
- A document $D = \{s_1, s_2, \ldots, s_n\}$
- Each sentence s_i is represented by a corresponding node $v_i \in V$



HEGEL

☐ Node Representation

- Adopt sentence-BERT as a sentence encoder to embed the semantic meanings of sentence as $X = \{x_1, x_2, \dots, x_n\}$
- Adopt the hierarchical position embedding
- Initial node representation



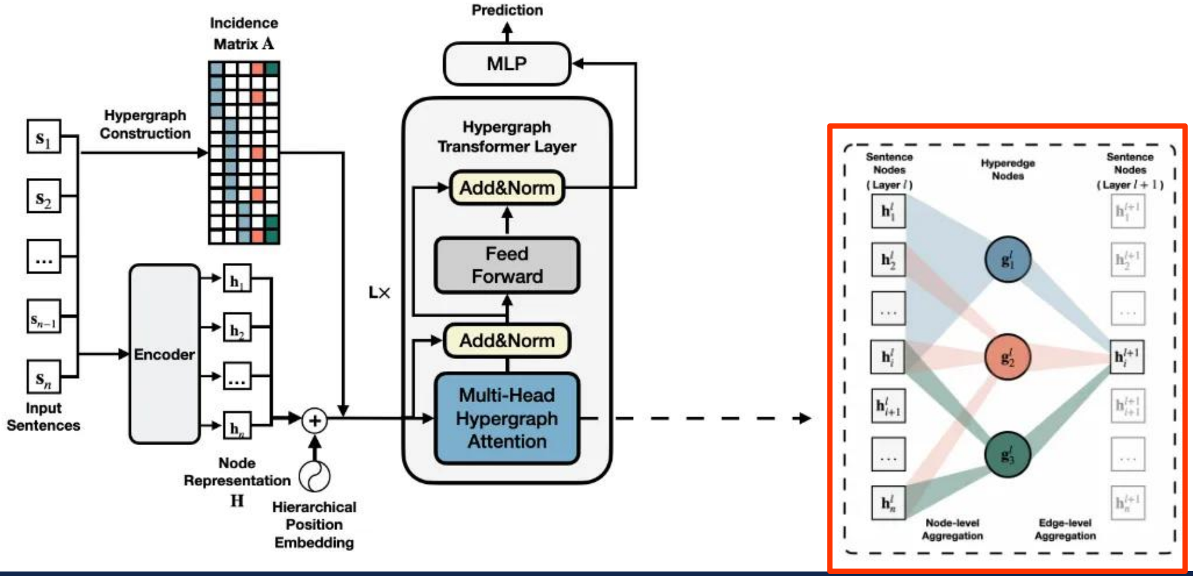
Hypergraph Construction

- Section Hyperedges
  - Sentences within the same section tend to have the same semantic focus
- Topic Hyperedges
  - Apply the Latent Dirichlet Allocation (LDA) to extract the latent topic relations
- Keyword Hyperedges
  - Extract keywords for academic papers with KeyBERT
- Fuse the three hyperedges by concatenation
  - $A = A_{sec} || A_{topic} || A_{kw}$

HEGEL

Hypergraph Transformer Layer

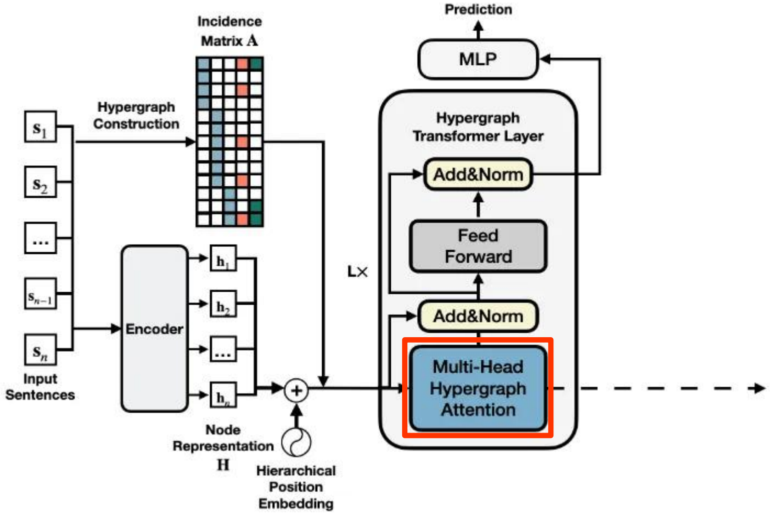
- Hypergraph Attention
  - Obtain all m hyperedge representations $\{g_1^l, g_2^l, \dots, g_m^l\}$
  - Update node representations H^{l-1} based on the updated hyperedge representations

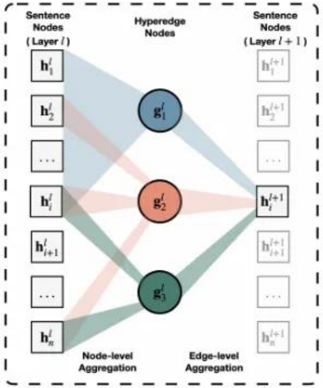


HEGEL

☐ Hypergraph Transformer Layer

- Multi-head Hypergraph Attention
  - ☐ $MH-HGA(H,A)=\sigma\left(W_O||_{i=1}^{h}head_i\right)$
  - ☐ $head_i=HGA(H,A)$

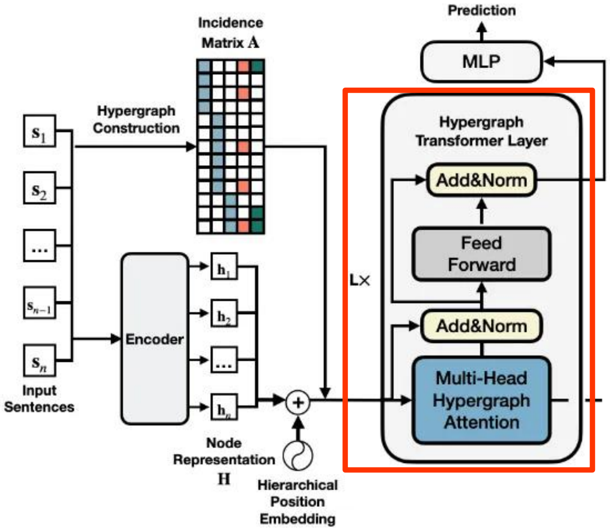


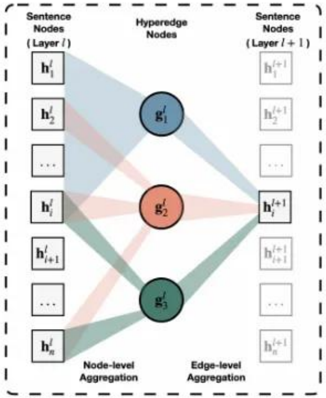


HEGEL

Hypergraph Transformer

- $H'^{(l)} = LN\left(MH - HGA\left(H^{l-1}, A\right) + H^{l-1}\right)$
- $H^{l} = LN(FFN\left(H'^{(l)}\right) + H'^{(l)}$





HEGEL

☐ Training Objective

- Compute the confidence score for selecting each sentence using a MLP
  - $z_i = LeakyReLU\left(W_{p1} h_i^L\right)$
  - $\hat{y}_i = sigmoid\left(W_{p2} z_i\right)$

- Optimize with binary cross-entropy loss
  - $L = -\frac{1}{N \cdot N_d} \sum_{d=1}^{N} \sum_{i=1}^{N_d} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$

Experiments

☐ Datasets

- Scientific paper summarization datasets

	Arxiv	PubMed
# train	201,427	112,291
# validation	6,431	6,402
# test	6,436	6,449
avg. document length	4,938	3,016
avg. summary length	203	220

Table 1: Statistics of PubMed and Arxiv datasets.

Experiments

Experimental Results on two benchmark datasets

- Outperform overall SOTA methods

Models	PubMed			ArXiv		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ORACLE	55.05	27.48	49.11	53.88	23.05	46.54
LEAD	35.63	12.28	25.17	33.66	8.94	22.19
LexRank (2004)	39.19	13.89	34.59	33.85	10.73	28.99
PACSUM (2019)	39.79	14.00	36.09	38.57	10.93	34.33
HIPORANK (2021)	43.58	17.00	39.31	39.34	12.56	34.89
Cheng&Lapata (2016)	43.89	18.53	30.17	42.24	15.97	27.88
SummaRuNNer (2016)	43.89	18.78	30.36	42.81	16.52	28.23
ExtSum-LG (2019)	44.85	19.70	31.43	43.62	17.36	29.14
SentCLF (2020)	45.01	19.91	41.16	34.01	8.71	30.41
SentPTR (2020)	43.30	17.92	39.47	42.32	15.63	38.06
ExtSum-LG + RdLoss (2021)	45.30	20.42	40.95	44.01	17.79	39.09
ExtSum-LG + MMR (2021)	45.39	20.37	40.99	43.87	17.50	38.97
HiStruct+ (2022)	46.59	20.39	42.11	45.22	17.67	**40.16**
PGN (2017)	35.86	10.22	29.69	32.06	9.04	25.16
DiscourseAware (2018)	38.93	15.37	35.21	35.80	11.05	31.80
TLM-I+E (2020)	42.13	16.27	39.21	41.62	14.69	38.03
DANCER-LSTM (2020)	44.09	17.69	40.27	41.87	15.92	37.61
DANCER-RUM (2020)	43.98	17.65	40.25	42.70	16.54	38.44
HEGEL (ours)	**47.13**	**21.00**	**42.18**	**46.41**	**18.17**	39.89

Table 2: Experimental Results on PubMed and Arxiv datasets.

Experiments

☐ Ablation Study

Model	ROUGE-1	ROUGE-2	ROUGE-L
full HEGEL	**47.13**	**21.00**	**42.18**
w/o Position	46.86	20.05	41.91
w/o Keyword	46.92	20.71	42.03
w/o Topic	46.35	20.30	41.48
w/o Section	45.63	19.30	40.71

Table 3: Ablation study results on PubMed dataset.

Experiments

Hyperedge Analysis

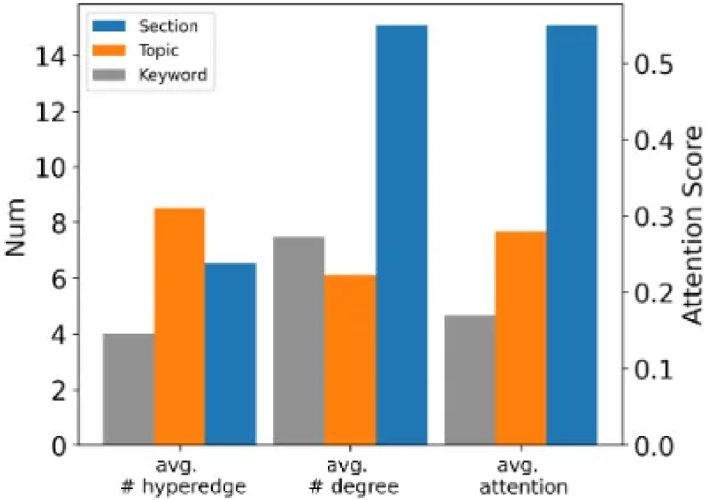


Figure 3: Average attention distribution over three types of hyperedges on PubMed dataset.

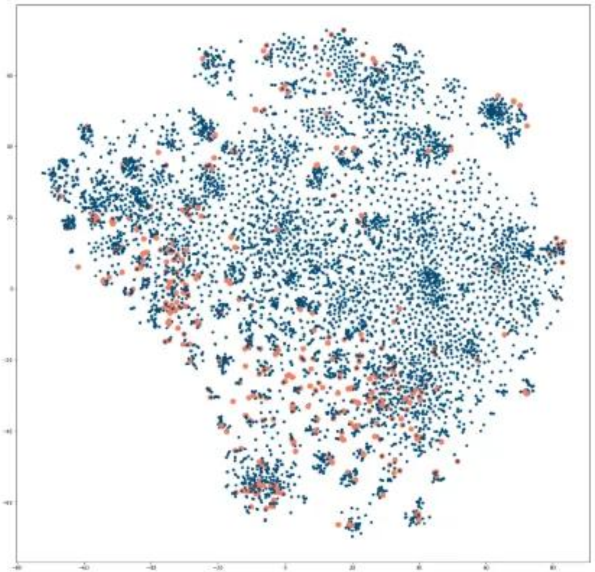


Figure 4: Visualization of sentence nodes embeddings for 100 documents in PubMed test set.

Conclusion

- **Long Document Summarization**
  - Different for sequential models to capture
- **GNN-based approaches**
  - Only model the pairwise interaction between sentences
  - Incapable of fusing sentence interactions from different perspectives
- **HEGEL**
  - Model high-order cross-sentence relations with hypergraphs
  - Use the attention mechanism
- **Experiments**
  - Outperform overall SOTA methods

Appendix

Node Representation

- Hierarchical position embedding
  - $HPE(s_i) = \gamma_1 PE(p_i^{sec}) + \gamma_2 PE(p_i^{sen})$
  - $PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}})$
  - $PE(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}})$

- Initial node representation $H^0 = \{h_1^0, h_2^0, \dots, h_n^0\}$
  - $h_i^0 = x_i + HPE(s_i)$

Appendix

☐ Hypergraph Transformer Layer

- Hypergraph Attention
  - ☐ Obtain all m hyperedge representations $\{g_1^l, g_2^l, \ldots, g_m^l\}$

$$\mathbf{u}_k = \mathrm{LeakyReLU}\left(\mathbf{W}_h \mathbf{h}_k^{l-1}\right), \quad \alpha_{jk} = \frac{\exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_k\right)}{\sum_{v_p \in e_j} \exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_p\right)}, \quad \mathbf{g}_j^l = \mathrm{LeakyReLU}\left(\sum_{v_k \in e_j} \alpha_{jk} \mathbf{W}_h \mathbf{h}_k^{l-1}\right)$$

  - ☐ Update node representations H^{l-1} based on the updated hyperedge representations

$$\mathbf{z}_k = \mathrm{LeakyReLU}\left(\left[\mathbf{W}_e \mathbf{g}_k^l \| \mathbf{W}_h \mathbf{h}_i^{l-1}\right]\right) \quad \beta_{ki} = \frac{\exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_k\right)}{\sum_{v_i \in e_q} \exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_i\right)} \quad \mathbf{h}_i^l = \mathrm{LeakyReLU}\left(\sum_{v_i \in e_k} \beta_{ij} \mathbf{W}_e \mathbf{g}_k^l\right)$$