How Can Graph Neural Networks Help Document Retrieval: A Case Study on CORD19 with Concept Map Generation

Hejie Cui, Jiaying Lu, Yao Ge, Carl Yang
(corresponding: j.carlyang@emory.edu)

Department of Computer Science, Emory University, Atlanta, GA 30322, USA

Contribution Highlights

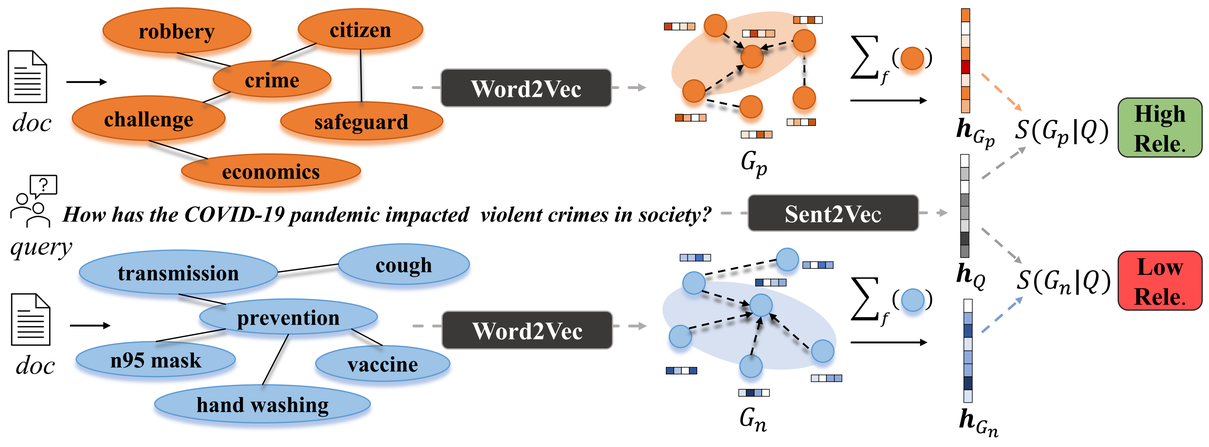


Figure: An overview of GNN-based document retrieval.

In this work, we explore how GNNs can help document retrieval with generated concept maps, consisting of:

- Use constituency parsing to construct semantically rich concept maps from documents and design quality evaluation towards document retrieval.
- Investigate two types of graph models for document retrieval: the *structure-oriented* complex GNNs and our proposed *semantics-oriented* graph functions.
- Compare the retrieval results from different graph models and provide insights towards GNN model design for textual retrieval.

Introduction

Background

- Concept map models texts as a graph with words/phrases as vertices and relations between them as edges.
- Empowered by the structured document representation of concept maps, it is intriguing to apply powerful GNNs for tasks like document classification and retrieval.

GNNs for Document Retrieval

Follow the common two-step practice for the large-scale document retrieval tasks:

- Step 1: initial retrieval on the whole corpus with full texts using BM25.
- Step 2: re-rank with GNN models: construct concept map $G = \{V, E\}$ for the top 100 candidate document and apply GNNs on each individual concept map, where node representation $\boldsymbol{h}_i \in \mathbb{R}^d$ is updated through neighborhood transformation and aggregation. The graph-level embedding $\boldsymbol{h}_G \in \mathbb{R}^d$ is summarized over all nodes with a read-out function.
- Given a triplet (Q, G_p, G_n) composed by a relevant document G_p and an irrelevant document G_n to the query Q, the triplet loss function:

$$L(Q, G_p, G_n) = \max\{S(G_n \mid Q) - S(G_p \mid Q) + margin, 0\},$$

where $S(G \mid Q) = \frac{\boldsymbol{h}_G \cdot \boldsymbol{h}_Q}{\|\boldsymbol{h}_G\| \|\boldsymbol{h}_Q\|}$, $\boldsymbol{h}_G$ is the learned graph representation and $\boldsymbol{h}_Q$ is the query representation from a pretrained model.

- Retrieval in the testing phrase: documents are ranked according to the learned relevance score $S(G \mid Q)$.

Concept Map Generation

- Concept map distill structured information hidden under unstructured text and represent it with a graph.
- Existing methods based on name entity recognition (NER) or relation extraction (RE) suffer from limited nodes and sparse edges, rely on significant training data and predefined entities and relation types.
- We propose to use POS-tagging and constituency parsing to increase node/edge coverage, thus bolstering the semantic richness of the generated concept maps for retrieval. The interactions among extracted nodes are constructed by sliding window.

GNN-based CP Representation

Type 1: Structure-oriented complex GNNs

- The discriminative power of structure-oriente complex GNNs stems from the 1-WL test for graph isomorphism.
- We adopt two state-of-the-art ones, Graph isomorphism network (GIN) and Graph attention network (GAT).

Type 2: Semantics-oriented permutation invariant graph functions

In contrast, we propose a series of semantics-oriented graph functions:

- *N-Pool*: independently process each single node v_i by multi-layer perceptions and then apply a read-out function to aggregate node embeddings $\boldsymbol{a}_i$ into the graph embedding $\boldsymbol{h}_G$, i.e.,

$$\boldsymbol{h}_G = \text{READOUT}\left(\{\text{MLP}(\boldsymbol{a}_i) \mid v_i \in V\}\right).$$

- *E-Pool*: the edge embedding of each edge $e_{ij} = (v_i, v_j)$ is obtained by concatenating the node embedding $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ on its two ends to encode first-order interactions, i.e.,

$$\boldsymbol{h}_G = \text{READOUT}\left(\{cat(\text{MLP}(\boldsymbol{a}_i), \text{MLP}(\boldsymbol{a}_j)) \mid e_{ij} \in E\}\right).$$

- *RW-Pool*: for each sampled random walk $p_i = (v_1, v_2, \ldots, v_m)$ that encode higher-order interactions among concepts, the embedding is computed by the sum of all node embeddings on it, i.e.,

$$\boldsymbol{h}_G = \text{READOUT}\left(\{sum(\text{MLP}(\boldsymbol{a}_1), \text{MLP}(\boldsymbol{a}_2), \ldots, \text{MLP}(\boldsymbol{a}_m)) \mid p_i \in P\}\right).$$

They preserve the *message passing* mechanism of complex GNNs while focusing on the basic semantic units and different level of interactions between them.

Experiment and Analysis (1/2)

I. Evaluation of Concept Maps

Table: The similarity of different concept map pairs.

Pair Type	# Pairs	NCR (%)	NCR+ (%)	ECR (%)	ECR+ (%)
Pos-Pos	762,084	4.96	19.19	0.60	0.78
Pos-Neg	1,518,617	4.12	11.75	0.39	0.52
(*t-score*)	-	(*187.041*)	(*487.078*)	(*83.569*)	(*105.034*)
Pos-BM	140,640	3.80	14.98	0.37	0.43
(*t-score*)	-	(*126.977*)	(*108.808*)	(*35.870*)	(*56.981*)

→ Concept maps can indicate query document relevance and provide additional discriminative signals based on the initial candidates.

Experiment and Analysis (2/2)

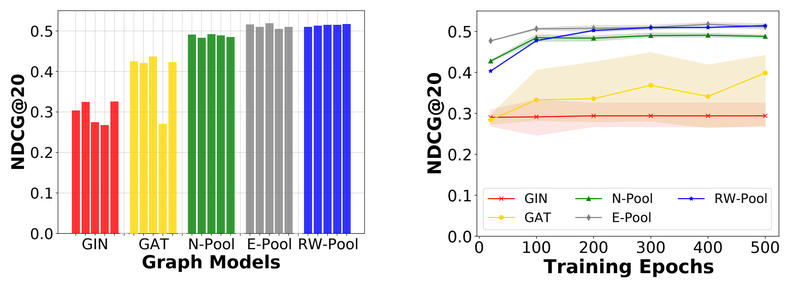
II. Retrieval Performance Results

Table: The retrieval performance of different models.

Type	Methods	Precision (%)		Recall (%)		NDCG (%)	
		k=10	*k*=20	*k*=10	*k*=20	*k*=10	*k*=20
Traditional	BM25	55.20	49.00	1.36	2.39	51.37	45.91
	Anserini	54.00	49.60	1.22	2.25	47.09	43.82
Structure-Oriented	GIN	35.24	34.36	0.77	1.50	30.59	29.91
	GAT	46.48	43.26	1.08	2.00	42.24	39.49
Semantics-Oriented	N-Pool	58.24	52.20	1.38	2.41	53.38	48.80
	E-Pool	59.60	53.88	1.40	2.49	56.11	51.16
	RW-Pool	**59.84**	**53.92**	**1.42**	**2.53**	**56.19**	**51.41**

- Structural-oriented GNNs fail to improve the baselines (BM25, Anserini).
- Semantics-oriented graph functions yield significant and consistent improvements over both baselines and structure-oriented GNNs.
- Demonstrate the potential of designing semantics-oriented GNNs for textual reasoning tasks such as classification, retrieval, etc.

III. Stability and Efficiency



(a) Stability comparison (b) Efficiency comparison

Figure: Stability and efficiency comparison of different graph models.

- Semantics-oriented functions perform more stable and improve efficiently during training.
- E-Pool and RW-Pool are consistently better than N-Pool, revealing the utility of simple graph structures.
- RW-Pool converges slower but achieves better and more stable results in the end, indicating the potential advantage of higher-order interactions.

Resources

Paper Code Poster