Topology-aware Retrieval Augmentation for Text Generation

Yu Wang
yuwang@uoregon.edu
University of Oregon
Eugene, OR, USA

Nedim Lipka
lipka@adobe.com
Adobe Research
San Jose, CA, USA

Ruiyi Zhang
ruizhang@adobe.com
Adobe Research
San Jose, CA, USA

Alexa Siu
asiu@adobe.com
Adobe Research
San Jose, CA, USA

Yuying Zhao
yuying.zhao@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Bo Ni
bo.ni@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Xin Wang
xin.wang.1@vanderbilt.com
Vanderbilt University
Nashville, TN, USA

Ryan Rossi
ryrossi@adobe.com
Adobe Research
San Jose, CA, USA

Tyler Derr
tyler.derr@vanderbilt.com
Vanderbilt University
Nashville, TN, USA

Abstract

Retrieval-augmented Generation (RAG) has been used to augment language models by retrieving texts from external databases. Since real-world texts are often connected in the graph (e.g., papers in citation networks), we use these relations to guide the retrieval process of RAG. Concretely, we investigate proximity and role-based relations, where the former considers topologically close nodes and the latter considers structurally similar nodes. We empirically verify their correlation to text relations, which motivates us to propose the framework of Topology-aware Retrieval-augmented Generation for text generation, which consists of a retrieval module to retrieve texts by their topological relations and an aggregation module to compose retrieved texts into prompts triggering LLMs for text generation. Extensive experiments verify the effectiveness of this framework, signifying the potential of equipping RAG with topological awareness. Our code is publically available at here.

CCS Concepts

• **Computing methodologies** → **Machine learning**.

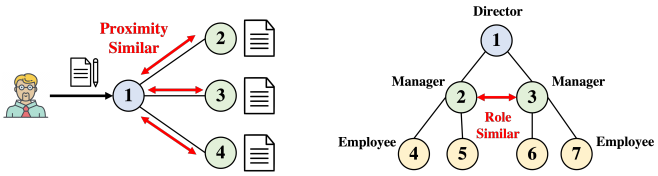
Keywords

Retrieval-Augmented Generation, Graph Structural Relations

ACM Reference Format:
Yu Wang, Nedim Lipka, Ruiyi Zhang, Alexa Siu, Yuying Zhao, Bo Ni, Xin Wang, Ryan Rossi, and Tyler Derr. 2024. Topology-aware Retrieval Augmentation for Text Generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679746

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679746



(a) Proximity-based Retrieval (b) Role-based Retrieval

Figure 1: Topology-aware Retrieval for Text Generation.(a): People write paper abstracts by referring to other papers cited in the related work.; (b): Employees owning the same local subgraph structures possess the same titles/responsibilities.

1 Introduction

Text generation focuses on creating human-readable texts based on the input text instructions [13, 16, 25, 56]. Despite the unprecedented success achieved by large language models (LLMs) in text generation [3, 54], their performance is still hampered by the limited knowledge available in input texts [63, 69, 74]. On one hand, relying solely on input text provides limited information misaligning with the abundant knowledge necessary for the desired output. On the other hand, while pre-training over vast corpora has equipped LLMs with world knowledge, this knowledge is encoded in their black-box-like parameters and there is no clear pathway to map these intriguing parameters to interpretable knowledge that can be faithfully used in text generation.

To address the issue of limited knowledge in input texts, Retrieval-augmented Generation (RAG) is used to create a well-informed context by retrieving knowledge from external databases [12, 27, 35, 41, 42, 53, 60, 61]. Unfortunately, most of these methods have been exclusively deployed for question-answering tasks [40, 53, 61], with a limited exploration in text generation tasks. More importantly, they neglect two fundamental knowledge in the topological pattern, proximity-based knowledge [20, 49, 75] (i.e., nodes that can be mutually reached via only few-hops walks) and role-based knowledge [1, 46, 48] (i.e., nodes with similar local subgraph structures). For example, in Figure 1(a), when writing the paper abstract (node 1), having access to its cited papers in its related work section (nodes 2-4) could greatly enhance writing efficiency, because these

referenced papers likely possess similar knowledge and narrative styles. Likewise, in Figure 1(b), employees holding comparable positions in a company, like managers 2 and 3, typically share similar job responsibilities and analogous communication styles in their emails. Understanding the email content and job responsibilities of one manager can provide insights into the role and communication style of the other manager.

Given the overlook of the above two topological relations in current RAGs, we study their potential in enhancing RAGs for text generation. To validate the feasibility of this idea, we address two key questions: can the text generation be improved by incorporating additional texts? To answer the first question, we incrementally increase the contextual similarity to the text to be generated and observe a corresponding performance increase. Secondly, can textual relations between any pair of nodes be reflected by their topological relations? To answer the second question, we analyze the correlation between the textual relations between any two nodes and their proximity/role-based topological relations. Affirmative answers to the above two questions inspire us to propose a framework, Topology-aware Retrieval-augmented Generation (Topo-RAG), which augments text generation by retrieving relevant texts based on their proximity/role-based topological similarity.

- **Bridging Topological and Textual Relations**: we discover the positive correlation between proximity/role-based topological relations of any pair of two nodes and their textual relations over nine datasets across four distinct domains, bridging the gap of node pairwise relations between topology space and text space.
- **Developing Topology-Informed RAG Framework**: we equip the RAG with topological awareness by retrieving texts from the graph database according to their proximity/role-based similarity to the target entity for which the text is being generated.
- **Comprehensive Empirical Analysis**: We construct a wide range of text-attributed graphs from diverse domains and conduct comprehensive experiments to verify the effectiveness of our framework. Moreover, we pioneer the use of node classification and link prediction to evaluate the quality of the generated texts.

2 Preliminary

2.1 Motivative Examples

To motivate the analysis of the correlation between the textual and topological relation, we take two examples, one analyzing the proximity-based relation (Figure 2(a)-(b)), and the other one analyzing the role-based relation (Figure 2(c)-(d)).

2.1.1 Textual Relation with Proximity-based Topological Relation. In Figure 2(a), we extract a subgraph centering around node 0 from the citation network, Cora, where each node represents a paper with the abstract as its textual information, and each edge indicates a reference relation between the corresponding two paper nodes. Meanwhile, we obtain textual embedding of each node by feeding its abstract through sentence-transformer [45] and calculate the pairwise cosine similarity in Figure 2(b). Comparing Figure 2(a) and (b), we observe that as nodes become further away from node 0, their textual similarity to node 0 gradually decreases, indicating the textual similarity between any pair of nodes is correlated to their proximity-based topological distance.

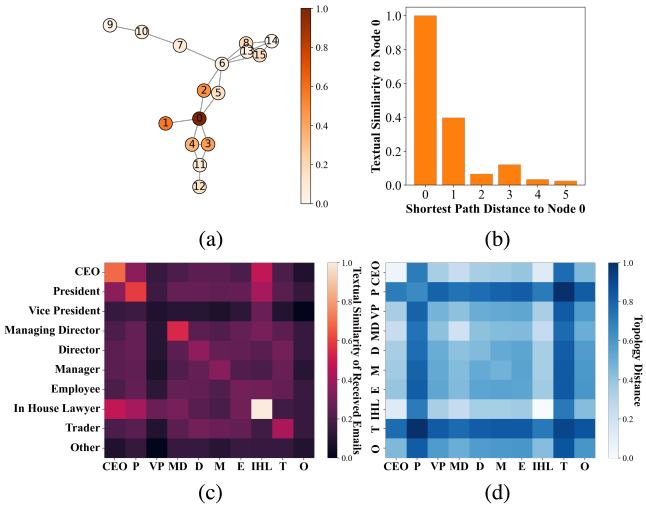


Figure 2: (a)-(b): Cora citation network where nodes are papers and edges are reference relations. Papers that are topologically closer to paper 0 have higher textual similarity to its text; (c)-(d): Eron-email network where nodes represent employees and edges denote their email communications. For each row in the two heatmaps, employees in diagonal entries share higher textual similarity and lower role-based topological distance than the ones belonging to off-diagonal entries in the same row. This indicates that employees with the same roles share a higher textual similarity and lower topological distance than employees with different roles.

2.1.2 Textual Relation with Role-based Topological Relation. We use the Eron-email dataset [7] where we have access to emails sent from/received by employees with their job titles. We feed each email through the sentence-transformer [45] to obtain each email embedding. Then, we calculate the embedding of each employee by averaging the embeddings of their received emails. After that, we calculate pairwise textual similarity among employees and group them based on their job titles in Figure 2(c). For each row, we observe that the diagonal entry generally has higher textual similarity compared with other off-diagonal entries in the same row, indicating higher content similarity of emails received among employees possessing the same role in a company.[1] Furthermore, we construct the graph with nodes being employees and edges being their email communications. We run GraphWave [8] to obtain employees' role-based embeddings and encode their local structural information. We calculate pairwise L2 distance among employees and group them based on their job titles. Similarly, we observe a generally lower role-based topological distance for employees possessing the same roles, indicating their local subgraph structures are aligned with their job titles in reality.

The above two observations motivate us to explore the potential of augmenting text generation by leveraging topology information. In the next, we introduce notations used throughout this paper and formulate the task of topology-aware text generation.

[1]Similar observation is also found when analyzing their sent emails in Figure 8 in Appendix A.1.

2.2 Notations

Let $\mathcal{S} = \{S_i\}_{i=1}^{m+n}$ be the set of $m+n$ text sequences where S_i represents the i^{th} sequence. Assume we also have access to an additional graph connecting these textual sequences $G = (\mathcal{V}, \mathcal{E})$ where each node v_i corresponds to a textual sequence S_i and the edge e_{ij} denotes the connection between node v_i and v_j. Furthermore, the adjacency matrix of this graph is notated as $\mathbf{A} \in \mathbb{R}^{(m+n)\times(m+n)}$ where $\mathbf{A}_{ij} = 1$ if there is an edge e_{ij} connecting v_i and v_j, and $\mathbf{A}_{ij} = 0$ otherwise. Let $\mathcal{N}_i$ be the neighbors of node v_i. We formulate the topology-aware text generation in the following.

2.3 Task Formulation

Given a set of textual sequences $\mathcal{S} = \mathcal{S}^{\text{Full}} \cup \mathcal{S}^{\text{Partial}}$ where $\mathcal{S}^{\text{Full}} = \{S_i\}_{i=1}^{n}$ comprises sequences with completely accessible texts, and $\mathcal{S}^{\text{Partial}} = \{S_i\}_{i=1}^{m}$ consists of sequences with texts that are only partially observable. As the objective of many text-generation applications is to generate complete texts based on the first few pre-existing words/sentences, the "partially observed texts" in this paper refer to the initial words provided in a sequence. Let the partially observed text be X_i for i^{th} sequence and its unobserved counterpart be Y_i. We aim to leverage LLMs $\mathcal{F}$ to generate the sequence $\widehat{Y}_i$ utilizing the information from input X_i, other fully observed texts in $\mathcal{S}^{\text{Full}}$ and their topological relations in G, with the expectation to recover the ground-truth sequence Y_i:

$$\widehat{Y}_i = \mathcal{F}(\Omega(X_i, \mathcal{S}^{\text{Full}}, G)), \ \forall i \in \{1, 2, ..., m\}. \tag{1}$$

Comparing with solely relying on the partially observed text X_i for the generation, i.e., $\widehat{Y}_i = \mathcal{F}(X_i)$, Eq. (1) leverages Ω to further retrieve the additional sequences from $\mathcal{S}^{\text{Full}}$ based on the topological knowledge in the graph structure G. The general hypothesis here is that for each pair of two textual sequences S_i, S_j and their corresponding nodes v_i, v_j in the graph, if 1) the text generation of LLMs can benefit from providing extra texts that are similar to the target text and 2) the textual similarity $\phi_{ij}^{\text{Text}} = \phi^{\text{Text}}(S_i, S_j)$ is correlated to their topological similarity $\phi_{ij}^{\text{Topo}} = \phi^{\text{Topo}}(v_i, v_j)$, then incorporating those topologically-similar sequences would benefit the generation of the current texts. Verifying the above hypothesis requires answering the following two questions:

- $\boldsymbol{Q_1}$: Would the text generation with a pre-trained large language model benefit from providing texts that have higher textual similarity to the current text to be generated?
- $\boldsymbol{Q_2}$: Is there any correlation between the textual similarity of two sequences and the topological similarity of their corresponding nodes in the graph?

Next, we address $\boldsymbol{Q_1}$ by empirically demonstrating the performance increase when providing additional texts with increasing textual similarity to the target text (Figure 3 in Section 3). We address $\boldsymbol{Q_2}$ by formally introducing two types of topological similarity, proximity-based one and role-based one, and analyzing their correlations to textual similarity (Figure 4/5 in Section 4). Successfully answering these two questions would motivate the proposal of our framework Topology-aware Retrieval-Augmented Generation (TopoRAG).

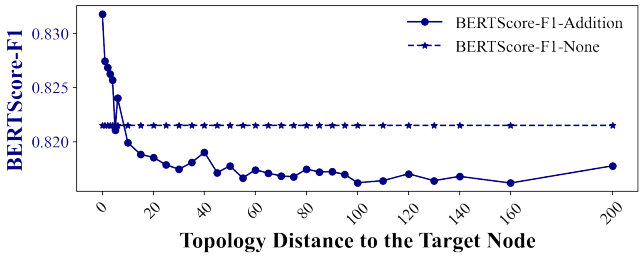


Figure 3: Comparing text generation between the scenario "Addition" where we include additional texts based on their proximity-based topological similarity to the target node and the scenario "None" where we include no additional texts but only based on its partially observed starting words.

3 Would text generation benefit from providing additional texts?

To address the first research question $\boldsymbol{Q_1}$, we compute the topological similarity between the target node designated for text generation and all other nodes using their proximity-based topological embedding according to Eq (3). Then, we rank all other nodes based on the topological similarity to the target node, iteratively select three consecutive nodes from the ranked list (ranging from positions k to $k+3$), and incorporate their texts as supplementary context to enhance the text generation. Figure 3 shows the generation performance of paper abstracts on Cora as we sequentially increase the topological rank of the selected additional nodes.

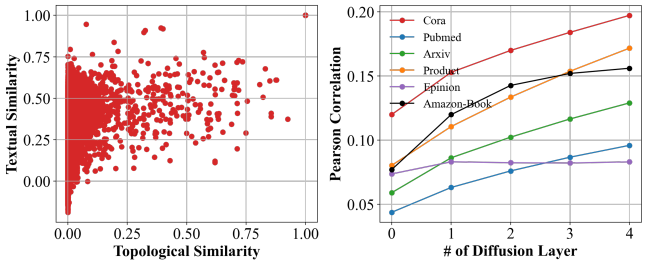
Firstly, compared with solely based on its partially observed starting words (horizontal line), the text generation performance is better when including nodes that are among the Top 6. This finding highlights the benefits of incorporating additional texts in augmenting the current text generation. Secondly, by gradually increasing the rank of nodes we select, the performance gradually decreases according to BertScore-F1, demonstrating that the benefit of including additional texts is correlated to the similarity of those texts to the target one, answering $\boldsymbol{Q_1}$.

4 Is topological similarity correlated to textual similarity?

Assuming the topological similarity ϕ^{Topo} and the textual similarity ϕ^{Text} are two random variables where their specific realizations correspond to their values for a specific pair of nodes, then their Pearson correlation is computed as:

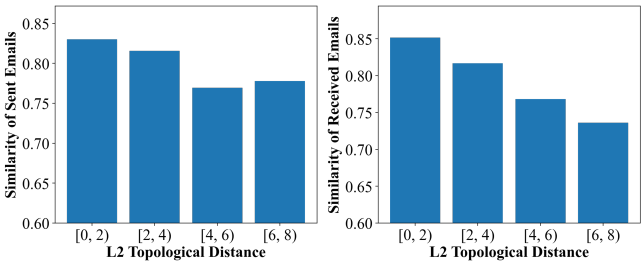
$$r = \frac{N^2 \sum_{i,j=1}^{N,N} \phi_{ij}^{\text{Text}} \phi_{ij}^{\text{Topo}} - \sum_{i,j=1}^{N,N} \phi_{ij}^{\text{Text}} \sum_{i,j=1}^{N,N} \phi_{ij}^{\text{Topo}}}{\sqrt{[N^2 \sum_{i,j=1}^{N,N} (\phi_{ij}^{\text{Text}})^2 - (\sum_{i,j=1}^{N,N} \phi_{ij}^{\text{Text}})^2][N^2 \sum_{i,j=1}^{N,N} (\phi_{ij}^{\text{Topo}})^2 - (\sum_{i,j=1}^{N,N} \phi_{ij}^{\text{Topo}})^2]}}, \tag{2}$$

where $\phi_{ij}^{\text{Text}} = \phi^{\text{Text}}(S_i, S_j)$ is defined as the semantic similarity and it is computed as the cosine similarity of textual embeddings from sentence-transformer. $\phi_{ij}^{\text{Topo}} = \phi^{\text{Topo}}(v_i, v_j)$ defines the topological similarity between two nodes v_i, v_j. Following previous works [8, 46], we measure this topological similarity between two nodes by the similarity of their topological embeddings. In the next, we explore two types of node topological embeddings: proximity-based and role-based ones.



(a) (b)

Figure 4: Correlation between Proximity-based Topological Similarity and Textual Similarity. (a): In Cora, as the topological distance between two paper nodes decreases, their textual similarity increases. (b): The Pearson correlations across different datasets are all positive and increase as the number of diffusion layer k in Eq. (3) increases.



(a) $r = -0.1937$ (b) $r = -0.3295$

Figure 5: Correlation between Role-based Topological Distance and Textual Similarity on Eron-Email Dataset where role-based topological distance is calculated based on the L2-distance between embeddings of two employees obtained from GraphWave while textual similarity is calculated as the average cosine similarity of textual embeddings of either the sent (a) or received emails (b) between two employees.

4.1 Proximity-based Topological Similarity

Proximity-based topological similarity quantifies the similarity between two nodes via their topological distance in the graph. The general intuition here is that two topologically close nodes usually have a higher textual similarity.

Conventional ways of characterizing the topological distance between two nodes in a graph include the shortest path distance and the shallow embeddings (e.g., random walk, personalized page rank and diffusion [15, 17]). In this work, we choose the diffusion-based one due to its thorough consideration of all potential paths in contributing to the proximity between any two nodes. Given a degree-normalized adjacency matrix $\widehat{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$ and an identity matrix $\mathbf{I} \in \{0,1\}^{N\times N}$ uniquely identifying each node, we perform diffusion to obtain the propagated node embeddings as:

$$\mathbf{P} = \sum_{k=1}^{K} \alpha_k \widehat{\mathbf{A}}^k \mathbf{I}, \tag{3}$$

where $\mathbf{P}_{ij}$ quantifies the influence of node v_j on node v_i considering paths of lengths varying from 1 to K since two nodes that are topologically close to each other should receive similar influence from all other nodes in this graph. However, directly performing this diffusion requires $O(KN^3)$ for time and $O(N^2)$ for space complexity. To make this computation scalable [5], we further project the unique node label matrix $\mathbf{I}$ via random gaussian projection by replacing $\mathbf{I} \in \mathbb{R}^{N\times N}$ with $\mathbf{R} \in \mathbb{R}^{N\times d} \sim \mathcal{N}(\mathbf{0}^d, \Sigma^d)$ with $d << N$, which effectively reduces the time/space complexity to $O(KN^2d)/O(Nd)$. Then the topological similarity between two nodes is computed as the cosine similarity between their corresponding propagated node embeddings, i.e., $\phi_{i,j}^{\text{Topo}} = \cos(\mathbf{P}_i, \mathbf{P}_j)$.

To verify the correlation between the proximity-based topological similarity and textual similarity, we conduct correlation analysis on datasets in Figure 4. On Cora dataset, we run diffusion according to Eq. (3), obtain node embeddings $\mathbf{P}$, calculate the pairwise topological similarity, and visualize it along with the pairwise textual similarity in Figure 4(a) [2]. We can see the pairwise textual similarity increases as the pairwise topological similarity increases. Moreover, we calculate the Pearson correlation across different datasets at different diffusion layers k in Eq. (3) in Figure 4(b). In all six datasets, the correlation is positive and increases as the diffusion layer increases as the higher diffusion layer considers the higher-order paths in quantifying the topological proximity, which becomes more aligned with their textual similarity. Different from Cora/Pubmed/Arxiv/Product/Amazon-Book, the correlation on the Epinion dataset does not increase since reviews posted by the same person may not necessarily be similar if the products are different.

[2]Similar analysis on other datasets are included in Figure 9 in Appendix A.2

4.2 Role-based Topological Similarity

Unlike proximity-based topological similarity which considers nodes residing closely in one network to be similar, role-based topological similarity focuses on identifying nodes with topologically similar neighborhoods [46]. Intuitively, nodes with similar local structures perform similar functions in the network and hence possess similar textual information in some aspects, such as the employees' tone should be different from the vice presidents' tone in a company [22]. Following this intuition, we employ one of the most representative embedding methods, GraphWave, that encodes the node local structure information in the next.

Following GraphWave [8], the spectral graph wavelet Ψ_a is calculated as:

$$\Psi_a = \mathbf{U}\text{Diag}(g_s(\lambda_1), \ldots, g_s(\lambda_N))\mathbf{U}^\top \mathbb{1}_a, \tag{4}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ and $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_N$($\mathbf{\Lambda} = \text{Diag}(\lambda_1, \lambda_2, ..., \lambda_N)$) is the eigenvalues of $\mathbf{L}$) with $g_s(\lambda) = e^{-\lambda s}$ being the kernel filter. $\mathbb{1}_a$ is the one-hot vector for node v_a. Ψ_a denotes the spectral graph wavelet for the heat kernel centered at node v_a and more specifically, its b^{th}-entry Ψ_{ba} denotes the amount of energy that node v_a has received from node v_b. Furthermore, GraphWave treats the spectral graph wavelet of node v_a, i.e., Ψ_a, as a probability distribution and characterizes it via the empirical characteristic function with i being the imaginary number here:

$$\phi_{\Psi_a}(t) = \mathbb{E}_{v_b \sim \mathcal{V}}(e^{it\Psi_{ab}}), \tag{5}$$

Eventually, structural embedding $\mathbf{P}_a$ of node v_a is obtained by sampling the 2-dimensional characteristics function at d evenly spaced points $t_1, t_2, ..., t_d$ and concatenating them as:

$$\mathbf{P}_a = ||_{t_1,t_2,...,t_d}[\mathrm{Re}(\phi_{\Psi_a}(t)), \mathrm{Im}(\phi_{\Psi_a}(t))]. \quad (6)$$

Then the topological similarity between two nodes is computed as the L2-distance between their corresponding structural embeddings, i.e., $\phi^{\mathrm{Topo}}_{a,b} = \frac{1}{||\mathbf{P}_a - \mathbf{P}_b||_2}$.

To verify the correlation between the role-based topological similarity and textual similarity, we conduct correlation analysis on the Eron-Email dataset in Figure 5. Following the experimental setting for Figure 2(c)-(d), we group each pair of employees based on their L2 topological distance and compute average textual similarity using their sent/received emails in Figure 5(a)/(b). We can see the negative correlation (-0.1937/-0.3295) between the role-based topological distance between two employees and the similarity of their sent/received emails.

To sum up, the positive responses to the preceding questions confirm two key insights. Firstly by the answer to Q_1, LLMs significantly benefit from incorporating additional texts during text generation. The closer the resemblance of these additional texts to the generated one, the greater the enhancement is in the output. Secondly, by the answer to Q_2, there is a positive correlation between textual and topological similarity. Drawing from these findings, we propose a novel framework using topological similarity to guide the retrieval of additional texts, thereby augmenting the quality of the generated text, i.e., Topology-aware Retrieval-Augmented Generation (Topo-RAG), which is introduced next.

5 Framework of Topo-RAG

Following Eq. (1), the retriever works by retrieving top-K nodes according to their topological similarity to the target text as follows:

$$\mathcal{T}_i = \Omega^{\mathrm{Retriever}}(X_i, \mathcal{S}^{\mathrm{Full}}, G) = \underset{v_j \in \mathcal{V}}{\arg\max}^{K} \phi^{\mathrm{Topo}}(v_i, v_j), \quad (7)$$

After that, we formulate the texts of nodes in $\mathcal{T}_i$ along with the partially observed text X_i of target node v_i into the prompt triggering LLM for text generation. Compared with conventional retrievals that only consider textual relations, the Topo-RAG framework considers topological relations. To implement this, it's necessary to pre-calculate the topological relations between every pair of nodes, a process requiring $O(|\mathcal{V}|^2)$ in both space and time complexity. To manage space constraints, we apply a top-k thresholding, reducing the space requirement significantly to $O(|\mathcal{V}|K)$. These computations are done in advance during an offline phase. Consequently, when a text generation request for a specific target node is made, we can retrieve the necessary information instantaneously from the pre-computed dictionary, maintaining a time complexity of $O(1)$. Note that prompting LLMs with the retrieved nodes of very long texts could exceed the input limits. Since this is a common issue for any RAG framework, one can equip our Topo-RAG framework with the existing strategies [28, 57] that handle this long-context issue. Here, we exclude instances where the context limit is exceeded. The length distribution analysis in Figure 8(b) reveals that the textual length of most nodes remains within acceptable limits, ensuring that our results and insights are still applicable to the original dataset even after the exclusion.

6 Experiments

Table 1: Statistics of Datasets. $\mathcal{S}^{\mathrm{Full}}$ denote nodes with fully available textual information while $\mathcal{S}^{\mathrm{Partial}}$ denote nodes with partially observable text.

Domain	Dataset	# Nodes	# Edges	# Instances ($\mathcal{S}^{\mathrm{Full}}/\mathcal{S}^{\mathrm{Partial}}$)	Splitting
Citation	Cora [6, 65]	2,708	5,429	2,522/186	Random
	Pubmed [6, 65]	19,717	44,335	17,786/1,931	Random
	Arxiv [6, 23]	16,316	53,519	14,791/1,525	Time
E-commerce	Product [6, 23]	16,475	60,015	15,790/685	Random
	Book [43]	7,252	203,438	6,526/726	Time
	Epinion [4, 73]	4,976	15,613	4,477/499	Time
	Music [43]	10,341	447,250	9,306/1,035	Time
	Pantry [43]	4,650	43,970	4,184/466	Time
Social	Eron-Email [52]	18,055	123,208	182,265/46,990	Random

6.1 Experimental Settings

6.1.1 Datasets. Although previous works [24, 33] borrow knowledge graphs to enhance text generation, the improvement mainly comes from the complex logical pattern encoded in the knowledge graph rather than proximity/structure topological patterns discussed in this paper. Therefore, we collect additional datasets to demonstrate the effectiveness of considering these two patterns in text generation, the details of which are discussed next:

- **Cora, Pubmed, Arxiv** [6, 23, 65]: Citation networks where nodes represent papers with abstracts as textual information, and edges signify reference relations. We divide nodes into fully-observed/partially-observed sets in a 90%/10% ratio and further remove nodes whose abstracts are less than 100 words. Then, we create the induced subgraph and remove edges connecting two nodes in the testing set to avoid information leakage. For the larger Arxiv network, due to resource constraints, we randomly select 2% nodes as seeds and apply the GraphSAGE sampling with the number of neighbors [2, 2] across two layers. Since Arxiv provides the publication time of each paper [23], we use the same preprocessing as Cora/Pubmed but follow the chronological order, imitating the real scenarios where users are writing papers with references to historical papers.
- **Book, Epinion, Music, Pantry** [4, 6, 23, 43, 73]: In digital e-commerce networks, each node represents a review and two reviews have an edge if they are either written by the same customer or posted on the same product. Considering the vast scale of the Book dataset, which encompasses 27,161,262 reviews, we only take the latest 1% reviews to construct the graph. We exclude any review whose length falls below 100 characters. We adhere to the same data splitting ratio used in Cora/Pubmed/Arxiv following the chronological order when the review was generated.
- **Products** [23]: Amazon product co-purchasing network where nodes represent products sold in Amazon and edges between two products indicate the co-purchase behaviors. Following Arxiv, we randomly select 0.1% among all product nodes as seeds and apply a GraphSAGE-based neighborhood sampling, with the number of neighbors as [2, 2] across two layers. Different from [23] using the sales ranking to split nodes, we randomly select 90%/10% nodes into fully-observed/partially-observed sets.

- **Eron-Email** [8]: In email communication networks, each node represents an employee in the company with his/her textual information being the historical written/received emails. We preprocess the original emails and extract the sender/receiver/text information of each email following script here[3].

6.1.2 Baselines. For baselines, as no previous works considered the proximity/role-based relations in RAG, we design baselines by equipping LLMs, GPT3 and GPT3.5 for text generation, with the following RAG strategies:

- **None**: we do not retrieve any additional texts but completely rely on the partially observed starting words.
- **Random (RD)**: we randomly retrieve K texts from the graph-structured knowledge base.
- **Text**: we calculate the semantic similarity between the partially observed texts in the target sequence and all other texts. Then we select the Top-K ones according to their semantic similarity.
- **Topo**: we calculate the topological similarity according to embeddings from Eq. (3)/(6) between the node of the target sequence and all other nodes. Then we rank them and select the top-K ones. This one is essentially our Topo-RAG framework.

6.1.3 Evaluation Tasks. To evaluate the quality of our generated text, we compare it with ground-truth text following established methodologies [36, 50]. We only focus on comparing the generated content, without considering any initially observed words. In addition, for the very first time, we introduce a task-oriented evaluation to assess the quality of the generated texts. The general motivation of this task-oriented evaluation is due to the growing relevance of text generation in real-world applications where generated content serves not only for direct use but also as input for downstream tasks. Specifically, we adopt two graph-based tasks, node classification and link prediction. In these two tasks, textual features of certain nodes are assumed to be reconstructed using various baselines including our Topo-RAG. With the reconstructed textual features of nodes after text generation, we then train graph machine learning models and evaluate their performance. We conduct the evaluation using GCN [32], SAGE [18], and MLP to exclude any model-induced bias during evaluation.

6.1.4 Evaluation Metrics. Following conventional works [36, 50], we use the BLEU-4/ROUGE-L/Bert-F1 score as the evaluation metrics to conduct a comprehensive analysis of the generated texts. In addition, we take the initiative to use the task-oriented evaluation, which quantifies the quality of generated texts based on whether they can fulfill purposes of downstream tasks, e.g., node classification and link prediction. For node classification, we report the average accuracy of testing nodes (in our setting, the testing nodes are assumed to be the ones with features to be reconstructed). For link prediction, following [23], we report the average Hits@100 of randomly selected edges.

6.1.5 Parameter Settings. For text generation, the number of retrieved texts and partially observed starting words are both set as 3. Moreover, we set the number of generated words to be 150, 250, 200, 150, 300, 500, 250, 200, 300 according to the average length of texts in Cora, Pubmed, Arxiv, Product, Book, Epinion, Music, Pantry, Eron-Email. For all datasets except Eron-Email, each node is only associated with one text sequence, hence we could directly calculate textual/topological similarity metric and retrieve the Top-3 accordingly. For Eron-Email where each node/employee possesses many texts/emails, we first query the sender and receiver of the target email to be generated and then collect emails from the two employees with the highest topological similarity to that sender and receiver. Furthermore, we select the Top-3 emails from those collected emails based on their textual similarity to the partially observed target text. For most of the hyperparameters used for evaluation with node classification and link prediction, we follow the same setting as [59] and [72]. In node classification, the hyperparameters are: training epoch is 1000, learning rate 0.01, weight decay 0.0005, early stopping 100, 2 layer graph convolution layer/MLP, dropout 0.5, number of hidden layers 64. In link prediction, the hyperparameters are: encoder learning rate 0.001, predictor learning rate 0.001, number of hidden layers 256, and dropout 0. We do not perform another dedicated hyperparameter search because the initial settings from [59, 72] have already undergone thorough hyperparameter optimization.

[3]https://github.com/mihir-m-gandhi/Enron-Email-Analysis/tree/main

6.2 Performance Comparison

6.2.1 Traditional Evaluation. Here we compare the text generation capabilities of GPT-3.5 and GPT-3 enhanced with our proposed Topo-RAG and other baseline methods. Due to resource constraints, we only randomly select 500 nodes along with their partially observed textual sequences to complete and report the average performance in Table 2. Overall, the proposed TopoRAG framework achieves the highest text generation performance with a significantly large margin, as shown by "Average", underscoring the benefits of integrating topological knowledge for retrieving additional contextual information in text generation. The second to best baseline is "Text" because it utilizes textual similarity to directly query relevant context, which augments the text generation to some extent. Interestingly, we also find that including random texts in the generation process, i.e., "RD", significantly improves performance compared to "None" which includes no additional texts at all. This improvement likely arises because each text within the same text-attributed network is generally related to the same domain, e.g., all texts in Cora are papers and all texts in Epinion are reviews. Hence, incorporating additional texts can provide insights into potential writing styles and usage of domain-specific terminology, which benefits the text generation of the target node. Comparing performance boosts across different evaluation metrics reveals that improvements under BLEU-4 often result in relatively larger gains compared to those under ROUGE-L and BERT-F1. This is because BLEU-4 emphasizes exact matches of words and phrases in the generated text against the reference, making it particularly sensitive to precise word matching. Conversely, ROUGE-L measures the overlap of n-grams between generated and reference texts, regardless of their order or precise phrasing, rendering it less sensitive than BLEU. BERT-F1, which evaluates semantic meaning, is even less sensitive to exact word matches, focusing more on the contextual alignment of the content. Overall, this suggests that incorporating additional context into text generation is more beneficial for achieving similar terminology or writing style rather than for capturing general meaning.

Table 2: Performance comparison of TopoRAG with baselines. The best results are in bold. BLEU is BLEU-4, ROUGE is ROUGE-L. Our TopoRAG almost achieves the best performance across all baselines on all datasets. "Average" is computed by averaging each metric across 9 datasets. "Boost" is computed by the relative performance gain from the second-to-best "Text" to the best "TopoRAG".

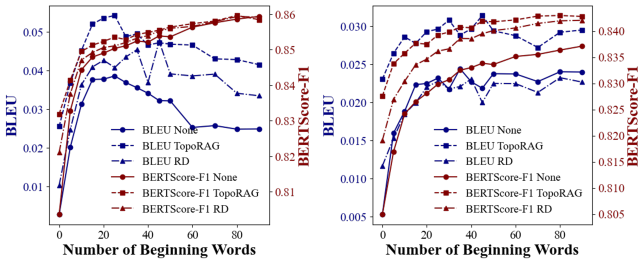
LLM	Retriever	Cora			Pubmed			Arxiv			Product			Book		
		BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1
GPT 3.5	**None**	1.46	15.90	82.15	1.52	14.63	80.28	1.22	15.25	81.87	1.57	15.09	81.98	1.05	14.75	80.98
	RD	1.78	16.42	83.10	2.38	15.83	81.32	2.64	16.22	83.03	1.65	14.90	82.09	1.34	14.77	82.30
	Text	1.77	16.43	83.05	2.27	15.49	81.37	2.23	15.89	82.96	2.44	15.50	82.16	1.77	15.30	82.53
	TopoRAG	**3.49**	**17.58**	**83.86**	**3.97**	**17.54**	**82.97**	**3.66**	**17.49**	**84.10**	**3.65**	**16.85**	**83.17**	**2.55**	**16.14**	**83.15**
	Boost	97.18%	7.00%	0.98%	74.89%	13.23%	1.97%	64.13%	10.07%	1.37%	49.59%	8.71%	1.23%	44.07%	5.49%	0.75%
GPT 3	**None**	1.16	15.40	81.07	1.19	14.10	79.42	0.88	14.47	80.53	1.90	14.71	80.85	0.75	14.20	80.03
	RD	1.72	16.45	82.79	2.48	15.88	81.11	2.50	16.44	82.90	1.48	14.50	81.16	1.00	15.10	82.18
	Text	2.21	16.50	82.85	2.32	15.69	81.24	2.17	15.82	82.60	2.93	15.54	81.24	1.15	15.30	81.83
	TopoRAG	**4.19**	**17.58**	**83.69**	**4.20**	**17.67**	**82.89**	**3.65**	**17.48**	**83.95**	**4.91**	**17.33**	**82.88**	**1.88**	**15.80**	**82.78**
	Boost	89.59%	6.55%	1.01%	81.03%	12.62%	2.03%	68.20%	10.49%	1.63%	67.58%	11.52%	2.02%	63.48%	3.27%	1.16%

LLM	Retriever	Epinion			Pantry			Eron-Email			Music			Average		
		BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1	BLEU	ROUGE	Bert-F1
GPT 3.5	**None**	0.47	10.46	78.80	0.99	14.63	81.23	2.40	9.18	79.27	1.33	14.07	80.37	1.33	13.77	80.77
	RD	0.62	10.89	80.34	1.25	15.16	82.69	2.06	9.71	80.33	1.85	14.17	81.70	1.73	14.23	81.88
	Text	0.59	10.75	80.29	1.79	15.40	82.68	3.09	10.60	79.92	1.24	14.20	81.63	1.91	14.40	81.84
	TopoRAG	**1.00**	**11.22**	**80.75**	**2.03**	**15.65**	**83.00**	**3.98**	**11.83**	**80.70**	**3.49**	**15.08**	**82.13**	**3.09**	**15.49**	**82.65**
	Boost	69.49%	4.37%	0.57%	13.41%	1.62%	0.39%	28.80%	11.60%	0.98%	181.5%	6.20%	0.61%	61.78%	7.57%	0.99%
GPT 3	**None**	0.34	10.68	78.42	0.72	13.57	80.03	1.65	7.62	78.42	1.19	13.70	79.43	1.09	13.16	79.80
	RD	0.48	10.86	80.16	1.04	14.98	82.49	3.30	10.55	79.49	1.17	14.25	81.53	1.69	14.33	81.53
	Text	0.38	10.85	79.54	1.64	15.28	82.72	3.65	**11.14**	79.79	4.03	15.54	81.81	2.28	14.63	81.51
	TopoRAG	**1.14**	**11.68**	**80.79**	**2.02**	**15.51**	**82.86**	**3.97**	10.69	**80.38**	**5.13**	**16.10**	**82.18**	**3.45**	**15.54**	**82.49**
	Boost	200.0%	7.65%	1.57%	23.17%	1.51%	0.17%	8.77%	-4.04%	0.74%	27.3%	3.6%	0.45%	51.32%	6.22%	1.2%

Table 3: Task-oriented Evaluation by comparing the node classification and link prediction performance of different baselines. The best results are in bold. NC - Node Classification; LP - Link Prediction.

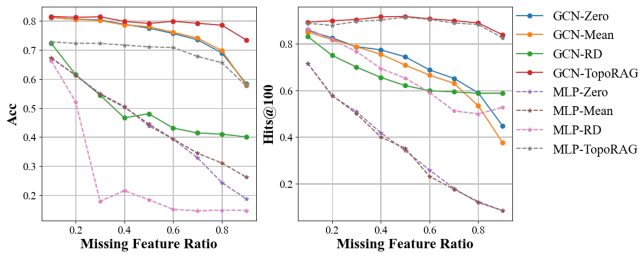
Model	Retriever	Cora		Pubmed	
		NC	LP	NC	LP
GCN	None	70.48±0.52	76.31±0.81	72.88±0.15	79.16±0.54
	RD	70.68±1.05	75.41±0.81	72.98±0.44	78.78±0.86
	Text	71.09±0.50	77.15±0.49	72.30±1.10	79.50±0.86
	TopoRAG	**75.49±0.26**	**89.12±0.49**	**77.80±0.53**	**81.33±0.40**
SAGE	None	60.75±1.34	80.44±0.98	64.24±3.47	79.77±0.62
	RD	57.20±1.69	78.61±0.58	64.76±2.43	80.00±0.72
	Text	57.07±2.95	82.40±0.61	64.18±2.28	80.48±0.15
	TopoRAG	**70.95±1.76**	**90.54±0.67**	**73.86±0.97**	**82.17±0.73**
MLP	None	42.03±0.38	73.85±0.92	49.58±0.15	77.43±0.51
	RD	39.93±0.41	70.34±0.96	49.08±0.50	76.66±0.56
	Text	47.57±0.73	72.82±0.83	51.32±0.51	77.63±0.63
	TopoRAG	**68.36±0.55**	**89.40±0.57**	**72.22±2.86**	**79.38±0.53**

6.2.2 Task-oriented Evaluation. In addition to conventional metrics, we also utilize task-oriented metrics to assess the quality of the generated texts. Specifically, we evaluate the performance of node classification and link prediction using the generated texts from different baselines. As shown in Table 3, TopoRAG consistently achieves the highest performance across all GNN backbones in both node classification and link prediction on the Cora and Pubmed datasets. This indicates that the texts generated by TopoRAG are more closely aligned with the main topics (node classification) and citations (link prediction) of their corresponding papers. Moreover, we observe that the performance improvement is even more pronounced when using MLP compared to GCN/SAGE. This is because GNN-based models inherently leverage neighborhood information to enhance context and hence compromise the augmenting effect caused by incorporating additional knowledge by Topo-RAG. Despite this inherent benefit, equipping them with TopoRAG still boosts the performance as TopoRAG additionally considers the potential of incorporating texts of non-neighboring nodes while the message-passing of GCN/SAGE only considers neighboring texts. In addition, we can see the additional benefit of TopoRAG on Cora is more than the one on Pubmed. We hypothesize that this is due to the higher correlation between textual similarity and topological similarity of Cora (0.2099) than the one of Pubmed (0.1039) in Figure 9. Notably, the superiority of our method is more pronounced in the results shown in Table 3 compared to "ROUGE" and "Bert-F1" in Table 2. This suggests that employing complex, task-oriented evaluation metrics can reveal subtle distinctions in the quality of generated texts. Such an approach is particularly valuable in the current landscape of Large Language Models (LLMs), providing a nuanced means to quantify the effectiveness of text generation.



(a) Cora (b) Book

Figure 6: As the number of beginning words increases, the BLEU score first increases and then decreases on Cora while the BertScore-F1 continually increases. TopoRAG Consistently achieves the highest text generation performance than the other two baselines.



(a) Cora-NC (b) Cora-LP

Figure 7: Performance of node classification (a) and link prediction (b) with varying rate of missing features on Cora dataset. Different baselines correspond to different feature imputation baselines.

6.3 Impact of Starting Words

We explore the impact of increasing the number of starting words, ranging from 0 to 100, on the performance of text generation for the Cora/Book datasets. Our findings reveal that TopoRAG consistently outperforms other methods. This advantage becomes even more obvious when the number of initial words becomes less. This trend suggests that a reduced count of starting words offers less contextual information, thereby heightening the need for the additional information provided by the TopoRAG. Moreover, we observe an interesting trend with BLEU scores initially increasing and then decreasing on Cora, whereas BertScore-F1 shows a consistent upward trajectory. This pattern can be attributed to the inherent characteristics of these evaluation metrics. The BLEU score focuses on the overlap of exact words. As we provide more starting words, the length of the remaining part of the target sentence decreases. Consequently, in the latter stages, the likelihood that the generated words precisely match the few remaining words diminishes, leading to a drop in BLEU scores. On the other hand, BertScore-F1 evaluates semantic embedding matching, which does not rely on exact word overlap. Therefore, as the number of provided starting words increases, LLMs gain a better understanding of the general context of the target text. This enhanced contextual understanding facilitates the generation of text that is semantically more aligned, explaining the consistent improvement in BertScore-F1. The reason why we do not see this first-increasing and then-decreasing trend with BLEU score on Book is due to the generally longer lengths of their texts compared with Cora, as verified in Figure 8(b).

6.4 Feature Imputation with TopoRAG

Many machine learning models assume a fully observed feature matrix. However, in practice, each feature is only observed for a subset of nodes due to constraints like privacy concerns or limited resources for data annotation [67]. In all these scenarios, the missing feature issues could catastrophically compromise the capability of machine learning models [47], which motivates many previous works developing solutions to handling missing feature issue [68].

Since our proposed TopoRAG can naturally generate node features in graph-based datasets, in this section, we evaluate its effectiveness in handling missing features by comparing its performance against other conventional baselines handling missing node features for graph-based tasks. We take the Cora dataset and consider two tasks, node classification (NC) and link prediction (LP). For NC, we follow the traditional semi-supervised setting [59] and for LP, we divide training/validation/testing links by 70%/10%/20% following [72]. For baselines handling missing feature issues, we consider baselines that set missing features to 0 **(Zero)**; a random value from a standard Gaussian **(Random)**; and also the global mean of that feature over the graph **(Global Mean)**. We consider equipping the backbone GCN and MLP with each of these strategies. In Figure 7, we visualize the performance of different baselines at different missing feature rates from 0.1 to 0.9. We can observe that TopoRAG consistently outperforms other strategies in both node classification and link prediction across all rates of missing features. This underscores the benefits of incorporating additional context in handling missing feature issues on graphs.

7 Related Work

7.1 Retrieval Augmented Generation

Retrieval augmented generation (RAG) refers to augmenting the performance of generative-based tasks via retrieving relevant information from external knowledge bases [9, 12, 21, 26, 31, 35, 37]. Conventionally, RAG is widely used in enhancing the performance of question-answering tasks by retrieving supporting facts including the answer to the given question [29, 31, 39, 64]. With the advent of LLMs, RAG gained more proliferation due to its capability to remedy the disadvantages of LLMs, such as mitigating the hallucination issue [2, 66, 71], enhancing interpretability [11], and enabling dynamic knowledge evolution of LLMs [38, 61]. Significant research efforts have recently been devoted to improving RAG through developing better retrieval methods [55, 58, 61, 70] or incorporating knowledge bases of various modalities [30, 34]. Following the former research trend, we enhance RAG retrieval methods by equipping it with topology awareness. Different from KG-based RAG where topology information is incorporated by retrieving triples from subgraphs around entities mentioned in the question [61], we explicitly consider proximity and role-based topological relations in guiding the retrieval, the related works of which are reviewed next.

7.2 Proximity/Role-based Topological Relations

Real-world entities often exhibit interconnected relationships that can be classified into two primary categories: proximity-based and structural-based relationships [1, 10, 48, 49, 51]. Proximity-based relationships between two nodes focus on their topological proximity, such as friends/relatives in social networks, co-cited academic papers in citation networks, and products co-purchased by the same customer [1, 14, 19, 75]. On the other hand, structural-based relations focus on the topological similarity between the local substructures of two nodes, e.g., two employees possessing the same title in a company share similar job responsibilities or airports acting as hubs following specific airline patterns [8, 46]. Previous works design various embedding-based methods capturing these two topological patterns. Methods such as Node2Vec and DeepWalk are mainly designed for capturing proximity-based topology patterns [17, 44] while Struc2Vec [46] and GraphWave [8] are mainly to capture structure-based patterns. Different from them, we explore whether these topological patterns also correlate to the textual patterns, e.g., whether two closely interacted people share similar tweet contents in their Twitter accounts or two structurally similar employees share similar job descriptions. Furthermore, we leverage the discovered correlation to guide the retrieval and enhance text generation.

8 Conclusion and Future Work

Given the limited knowledge provided in the input texts and the hallucination problem of LLMs, traditional approaches have leveraged RAG to incorporate extra knowledge. However, they predominantly focus on question-answering tasks with no significant investment in text-generation tasks. Furthermore, they overlook two critical types of knowledge embedded in the topological space, proximity-based and role-based knowledge. Therefore, this research aims to improve text generation performance by incorporating these two topological knowledge. Our empirical analysis reveals that LLMs benefit from additional texts that are similar to the target text. Moreover, by analyzing a wide range of text-attributed networks from diverse domains, we empirically verify the noticeable positive correlation between textual and proximity/role-based similarity. These findings have inspired us to develop Topology-aware Retrieval-Augmented Generation (Topo-RAG), a framework that enhances text generation by retrieving texts based on their topological similarities to the target text. We conduct comprehensive experiments to validate the effectiveness of Topo-RAG in text generation. Moreover, we take the initiative in utilizing node classification and link prediction to quantify the quality of the generated texts in a novel task-oriented manner. Additionally, we showcase an application of Topo-RAG in addressing missing feature issues in graph machine learning tasks.

Recognizing the importance of not only considering the quantity but also the structure of input knowledge in text generation [62], future work will focus on optimizing input formats by leveraging topological signals for question-answering and text-generation tasks. Moreover, we plan to assess the robustness of the TopoRAG framework by exploring the potential of attacking/defending over graphs to compromise/strengthen the capability of LLMs in completing downstream tasks.

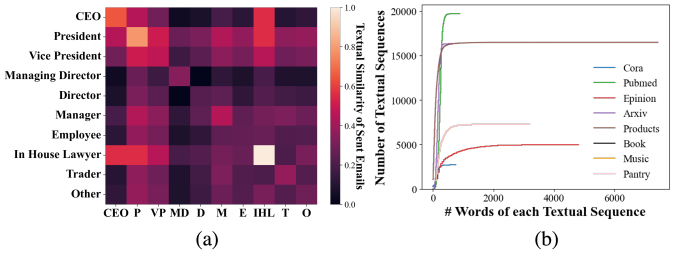


Figure 8: (a) The textual similarity of sent emails by pairs of employees grouped based on their job titles; (b) The distribution of passage length for each dataset.

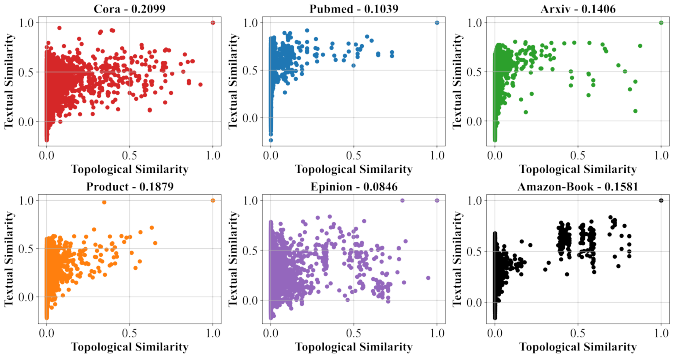


Figure 9: Correlation analysis between the textual similarity and proximity-based topological similarity over six datasets. The correlation shown beside the dataset name is positive across different datasets from different domains.

Acknowledgements

This research is supported by the National Science Foundation (NSF) under grant number IIS2239881 and Adobe Research.

A Appendix

In this section, we present supplementary findings that enhance the analysis from our primary study and further validate the generalizability of the observed phenomenon.

A.1 Textual Similarity over Sent Emails on Eron-Email Dataset

Following the same setting used for Figure 2(c), we visualize the textual similarity of emails sent by any pair of employees and further group them based on their job titles. Similar to the observation in Figure 2(c), we can see that people sharing the same job titles have similar textual patterns in their sent emails.

A.2 Additional Correlation Analysis

Here we conduct additional correlation analysis to demonstrate the positive correlation between proximity-based topological similarity and textual similarity. We can see a consistent positive correlation across six datasets from citation and E-commerce domains. This further justifies why TopoRAG achieves almost consistently higher performance than other baselines on all these datasets.

References

[1] Nesreen K Ahmed, Ryan Rossi, John Boaz Lee, Theodore L Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. 2018. Learning role-based graph embeddings. *arXiv preprint arXiv:1802.02896* (2018).

[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023* (2023).

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Chenwei Cai, Ruining He, and Julian McAuley. 2017. SPMC: Socially-aware personalized Markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497* (2017).

[5] Haochen Chen, Syed Fahad Sultan, Yingtao Tian, Muhao Chen, and Steven Skiena. 2019. Fast and accurate network embeddings via very sparse random projection. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 399–408.

[6] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2023. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393* (2023).

[7] Germán G Creamer, Salvatore J Stolfo, Mateo Creamer, Shlomo Hershkop, Ryan Rowe, et al. 2022. Discovering Organizational Hierarchy through a Corporate Ranking Algorithm: The Enron Case. *Complexity* 2022 (2022).

[8] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1320–1329.

[9] Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications. *arXiv preprint arXiv:2311.05876* (2023).

[10] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[11] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[12] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[13] Cristina Garbacea and Qiaozhu Mei. 2020. Neural language generation: Formulation, methods, and evaluation. *arXiv preprint arXiv:2007.15780* (2020).

[14] Alberto Garcia Duran and Mathias Niepert. 2017. Learning graph representations with embedding propagation. *Advances in neural information processing systems* 30 (2017).

[15] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. *Advances in neural information processing systems* 32 (2019).

[16] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.

[17] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[19] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[20] Haoyu Han, Xiaorui Liu, Feng Shi, MohamadAli Torkamani, Charu C Aggarwal, and Jiliang Tang. 2023. Towards Label Position Bias in Graph Neural Networks. *arXiv preprint arXiv:2305.15822* (2023).

[21] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *arXiv preprint arXiv:2402.07630* (2024).

[22] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1231–1239.

[23] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[24] Siyu Huo, Tengfei Ma, Jie Chen, Maria Chang, Lingfei Wu, and Michael J Witbrock. 2019. Graph enhanced cross-domain text-to-SQL generation. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*. 159–163.

[25] Touseef Iqbal and Shaima Qureshi. 2022. The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences* 34, 6 (2022), 2515–2528.

[26] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* (2022).

[27] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large Language Models on Graphs: A Comprehensive Survey. *arXiv preprint arXiv:2312.02783* (2023).

[28] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325* (2024).

[29] Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. 2022. Grape: Knowledge graph enhanced passage reader for open-domain question answering. *arXiv preprint arXiv:2210.02933* (2022).

[30] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*. PMLR, 15696–15707.

[31] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[32] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[33] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342* (2019).

[34] Rohan Kumar, Youngmin Kim, Sunitha Ravi, Haitian Sun, Christos Faloutsos, Ruslan Salakhutdinov, and Minji Yoon. 2023. Automatic Question-Answer Generation for Long-Tail Knowledge. (2023).

[35] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[36] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach LLMs to Personalize–An Approach inspired by Writing Education. *arXiv preprint arXiv:2308.07968* (2023).

[37] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110* (2022).

[38] Yuchao Li, Fuli Luo, Chuanqi Tan, Mengdi Wang, Songfang Huang, Shen Li, and Junjie Bai. 2022. Parameter-efficient sparsity for large language models fine-tuning. *arXiv preprint arXiv:2205.11005* (2022).

[39] Lihui Liu, Yuzhong Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. 2023. Knowledge Graph Question Answering with Ambiguous Query. In *Proceedings of the ACM Web Conference 2023*. 2477–2486.

[40] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. 2022. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1098–1108.

[41] Weimin Lyu, Xinyu Dong, Rachel Wong, Songzhu Zheng, Kayley Abell-Hart, Fusheng Wang, and Chao Chen. 2022. A multimodal transformer: Fusing clinical notes with structured EHR data for interpretable in-hospital mortality prediction. In *AMIA Annual Symposium Proceedings*, Vol. 2022. American Medical Informatics Association, 719.

[42] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553* (2020).

[43] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 188–197.

[44] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[45] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[46] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 385–394.

[47] Emanuele Rossi, Henry Kenlay, Maria I Gorinova, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. 2022. On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features. In

Learning on Graphs Conference. PMLR, 11–1.
[48] Ryan A Rossi and Nesreen K Ahmed. 2014. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2014), 1112–1131.
[49] Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. 2020. On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 5 (2020), 1–37.
[50] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *arXiv preprint arXiv:2304.11406* (2023).
[51] Satu Elisa Schaeffer. 2007. Graph clustering. *Computer science review* 1, 1 (2007), 27–64.
[52] Jitesh Shetty and Jafar Adibi. 2004. The Enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California* 4, 1 (2004), 120–128.
[53] Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2023. Graph neural prompting with large language models. *arXiv preprint arXiv:2309.15427* (2023).
[54] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
[55] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* (2022).
[56] Hao Wang, Bin Guo, Wei Wu, and Zhiwen Yu. 2020. Towards information-rich, logical text generation with knowledge-enhanced neural models. *arXiv preprint arXiv:2003.00814* (2020).
[57] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2024. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems* 36 (2024).
[58] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761* (2023).
[59] Yu Wang and Tyler Derr. 2021. Tree decomposed graph neural network. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 2040–2049.
[60] Yu Wang, Amin Javari, Janani Balaji, Walid Shalaby, Tyler Derr, and Xiquan Cui. 2024. Knowledge Graph-based Session Recommendation with Session-Adaptive Propagation. In *Companion Proceedings of the ACM on Web Conference 2024*. 264–273.
[61] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2023. Knowledge graph prompting for multi-document question answering. *arXiv preprint arXiv:2308.11730* (2023).
[62] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).
[63] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.
[64] Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756* (2020).
[65] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.
[66] Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469* (2023).
[67] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*. PMLR, 5689–5698.
[68] Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. 2020. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 19075–19087.
[69] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *Comput. Surveys* 54, 11s (2022), 1–38.
[70] Jiawei Zhang. 2023. Graph-ToolFormer: To Empower LLMs with Graph Reasoning Ability via Prompt Augmented by ChatGPT. *arXiv preprint arXiv:2304.11116* (2023).
[71] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* (2023).
[72] Tong Zhao, Gang Liu, Daheng Wang, Wenhao Yu, and Meng Jiang. 2022. Learning from counterfactual links for link prediction. In *International Conference on Machine Learning*. PMLR, 26911–26926.
[73] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 821–830.
[74] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention.. In *IJCAI*. 4623–4629.
[75] Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra. 2021. Node proximity is all you need: Unified structural and positional node and graph embedding. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 163–171.