ContextFlowGNN: A Novel Graph Neural Network for Dynamic Contextual Flow Analysis in NLP

Bikki kumar[1*]
bikkigupta378@gmail.com

Amrendra Singh[2]
singhamrendra1299@gmail.com

Aditya Kanaujiya[3]
adityakanaujiya600@gmail.com

Aanjneya Nayak[4]
aanjneya2@gmail.com

Aryan Singh[5]
aryansingh20091@gmail.com

Aditya Singh Sikarwar[6]
aditya20050202@gmail.com

Department *of* CSE(AI)
Krishna Institute of Engineering & Technology (KIET), Ghaziabad, Delhi-NCR, Uttar Pradesh, India

Abstract. Discourse coherence prediction, essential for automated essay scoring, dialogue systems, and multi-document summarization, is hindered by the inability of existing Graph Neural Network (GNN)-based Natural Language Processing (NLP) models to capture dynamic, multi-granular contextual dependencies. We propose ContextFlowGNN, a pioneering GNN framework that constructs a dynamic Context Flow Graph (CFG) integrating tokens, phrases, and discourse segments, enhanced by a physics-inspired flow-based attention mechanism, adaptive graph rewiring, hierarchical flow regularization, cross-granular message passing, temporal context decay, semantic flow modulation, discourse-aware node clustering, and attention guided edge pruning. ContextFlowGNN outperforms BERT with a significant increase in accuracy and a decrease in MSE. ContextFlowGNN demonstrates an accuracy improvement of 12.4% and a 31.7% drop in MSE as compared to BERT over curated dataset of 20000 essays, 10000 Reddit comments and 50000 news articles. Our extensive set of experiments includes ablation studies, cross-dataset experiments, error analysis, and qualitative visualizations. The datasets and code have been made publicly available.

Keywords: Contextual Flow, Discourse Coherence, Dynamic Graphs, Graph Neural Networks, Natural Language Processing.

1 Introduction

The coherence prediction refers to the understanding of the successive portions of text that should follow in a particular task. This task can make a model also learn in which order a sentence or a part of a sentence should occur and how well they sequence with each other. The arrangement of ideas must be clear and coherent to be effective. But

S. Sharma et al. (eds.), *Proceedings of the International Conference on Recent Advances in Intelligent and Sustainable Technologies (RAIST 2026)*, Atlantis Highlights in Intelligent Systems 17,
https://doi.org/10.2991/978-94-6239-707-1_23

existing NLP models, specifically transformer based architectures like BERT [1] and static GNN frameworks [2], are incapable of modelling dynamic, multi-granular (token, phrase, discourse) contextual dependencies. Due to sequential attention, transformers lack the capacity to model hierarchical relationships, whereas static GNNs cannot respond to changing contexts in long form texts. As a result, both fall short of performance in coherence-sensitive tasks. Because of its implications, this gap is critical. Within education, poor coherence prediction impacts the accuracy of automatic grading in feedback to students. In conversational AI, it hinders dialogue quality which affects satisfaction levels. In regards to news articles, it complicates the narrative tracking and contract or case coherence evaluation in legal analysis. This problem needs a new GNN framework which captures the multi-granular dependencies dynamically; one with superior performance, interpretability, and scalability for deployment in the wild in cross-disciplinary domains. Our proposed solution, ContextFlowGNN, is a novel GNN framework which builds a Context Flow Graph (CFG) which uses tokens, phrases, and discourse as nodes, and links them using edges whose weights use a flow-based attention mechanism inspired by physics. ContextFlowGNN incorporates adaptive graph rewiring, hierarchical flow regularization, cross-granular message passing, temporal context decay, semantic flow modulation, discourse-aware node clustering, and attention-guided edge pruning can be used to model a document as a flow of context or a fluid-like substance across various document units. Developed to enhance coherence prediction in discussions, it also facilitates the congruence of narratives across various documents and finds application in educational, journalistic, legal and other settings.
Our contributions are:

- The first-ever dynamic multi-granular Context Flow Graph (CFG), which connects and relates tokens, phrases and discourse segments with context-adaptive edges, in contrast to the static or single-level graph of prior work [2, 18, 22].
- A flow-based attention mechanism that draws inspiration from physics and introduces novel components, including temporal context decay, semantic flow modulation, and hierarchical flow regularization, to produce a smoothly continuous flow for context propagation, leveraging the analogy between context propagation and fluid dynamics in the vicinity of the time. This is the first application of such an approach for discourse coherence modelling in a NLP setting.
- A full suite of discourse-sensitive adaptations (discourse-aware node clustering, attention-guided edge pruning, cross-granular message passing, adaptive graph rewiring), complete with ablation studies demonstrating the contribution of each component.
- We released new curated coherence benchmarks, including 20000 essays ,10000 Reddit comments, and 50000 news articles, and we gauge strong performance gains over BERT, including 12.4% accuracy improvement and 31.7% MSE reduction (with error analysis and qualitative CFG visualizations).

2 Related work

Graph Neural Networks (GNNs) have done a game-changing impact in NLP, allowing them to perform tasks like text classification [2], relation extraction [3], question answering [4], abstractive summarization [5], and discourse parsing [6]. Graph Convolutional Networks (GCNs) [2] and GraphSAGE [7] utilize static graphs (e.g., dependency trees, knowledge graphs) for aggregation, while Graph Attention Networks (GAT) [8] rely on attention to determine the contribution of neighbors. The former offer good performance gains on a semantic role labeling, sentiment analysis, and text generationz[9-11]. Dynamic graph models [12] can tackle the temporal evolution issue in some dynamic domains such as social networks, and heterogeneous GNNs [13] can manage the complex multi-relational data. However, these two approaches are not very suitable for the rapidly evolving and dynamic context requirements in NLP. Researchers have previously investigated discourse coherence using entity-based models [14], centering theory [15], RNNs [16], LSTMs [17] and coherence graphs [18]. Transformer-based models like BERT [1], RoBERTa [19], and T5 [20] provide contextual embeddings but lack explicit graph structures for hierarchical relationships. Recent GNN-NLP advancements include DeepNote-GNN for clinical texts [21], NLA-GNN for non-local aggregation [22], graph-enhanced translation [23], and knowledge graph-augmented classification [24]. However, these focus on static graphs and tasks like classification or translation, not coherence prediction. Differing in their goals are graph-based summarization [25], coherence-aware dialogue systems [26], legal document analysis [27] which do not deal with dynamic, multi-granular structures or physics inspired attention. ContextFlowGNN offers a dynamic context-free grammar (CFG), unique attention mechanisms, and a coherence prediction specifically designed for graphs.

3 Methodology

3.1 Context Flow Graph (CFG) Construction

The CFG is a graph G = (V, E, W), where:

- **Nodes (V):** There are three sets of features, which consist of tokens, phrase, and discourse segments. Tokens consist of BERT embeddings with 768 dimensions. Phrases consist of noun/verb phrases using spaCy constituency parsing [28]. Discourse segments are sentences/paragraphs using Penn Discourse TreeBank annotations [29].
- **Edges (E):** Intra-granular (token-to-token within $|i-j| < 3$, phrase-to-phrase with $sim(hi , hj) > 0.5$) and inter-granular (token-to-phrase, phrase-to-discourse via containment).
- **Weights (W):** Dynamic, computed via flow-based attention and refined by pruning.

Construction steps:

- Use BERT tokenizer [1] to tokenize text and extract embedding. $H_i \in R^{768}$.
- Utilize spaCy's constituency parser for parsing phrases [28] and employ sentence segmentation or PDTB to analyze discourse segments [29].
- Develop edges within a granule according to syntactic proximity and semantic similarity
- Create inter-granular edges using containment relations (e.g., tokens in phrases).
- Begin with flow-based attention to initialize the edge weights, followed by refining and clustering and pruning.

The flow-based attention models context propagation as a fluid-like flow:

$$F_{ij} = \alpha \cdot sim(v_i, v_j) \cdot e^{\{-\beta . d_{ij}\}} \cdot e^{\{-\gamma . t_{ij}\}} \cdot \phi_{ij} \tag{1}$$

where:

$$sim(v_t, v_j) = \frac{h_i . h_j}{|| h_i|| \, ||h_j||} \tag{2}$$

$d_{ij} = ||h_i - h_j||$ (Euclidean distance),t_{ij} (temporal distance, e.g., token position difference) ,ϕ_{ij} (semantic flow modulation) and α, β, γ (learnable, initially 1.0, 0.1, 0.05). The temporal context decay (γ) models context degradation over long spans, enhancing coherence prediction.We introduce semantic flow modulation to adjust edge weights based on discourse roles:

$$\phi_{ij} = \sigma\left(W_{sem} \cdot [h_i;\ h_j] + b_{sem}\right) \tag{3}$$

Where $[h_i;\ h_j]$ concatenates node embeddings, $W_{sem} \in R^{1\times 1536}$, and σ is sigmoid. This emphasizes edges critical to discourse structure (e.g., argumentative connections).To enhance node representations, we apply discourse-aware node clustering:

$$C_i = argmin_k ||h_i - \mu_k||^2 \tag{4}$$

where μ_k are cluster centroids (k=5, initialized via k-means on BERT embeddings). Cluster assignments guide edge weight updates, aligning nodes with similar discourse roles.Edges with low attention scores are pruned:

$$F_{ij} < \theta \cdot \dot{F} \tag{5}$$

where θ = 0.1, and $\dot{F}$ is the mean weight. New edges are added based on:

$$sim'(v_i, v_j) = \frac{h_i^l \cdot h_j^l}{||h_i^l|| \, ||h_j^l||} \tag{6}$$

Edges are dynamically rewired by pruning low-weight edges and adding high-similarity edges, ensuring the CFG adapts to learned contexts.A regularization term stabilizes weights with λ = 0.01:

$$L_{reg} = \lambda \sum_{\{i,j \in E\}} (F_{ij} - \bar{F})^{\wedge}2 \tag{7}$$

Prioritizes inter-granular edges with η = 0.5:

$$h_i^{(l+1)} = \sigma\left(\begin{array}{c} \eta \sum_{j \in N_{intra}(i)} (F_{ij} W_{intra}^l h_j^l \ \ ) \\ + (1-\eta) \sum_{j \in N_{intra}(i)} F_{ij} W_{intra}^l h_j^l \end{array} \right) \tag{8}$$

The ContextFlowGNN framework evaluates the coherence of text documents by transforming linear text of documents into a multi-granular graph where information flow represents the logical transition of the text (i.e. word transition, sentence transition, paragraph transition, documentary transition). The input layer uses a pre-trained BERT-base to encode the sentences into a 768-dimensional contextual embedding. Doing so allows the BERT model to understand the context of the input sentences better. The first discourse clustering layer organizes the sentence nodes into cluster architectures on the basis of theme. This constructs a topology that demarcates local sentence-to-sentence transitions from global transitions which are topic-to-topic transitions. An Edge Flow Computation Module within this architecture calculates connection weights based on similarity, time ordering, and structure level.

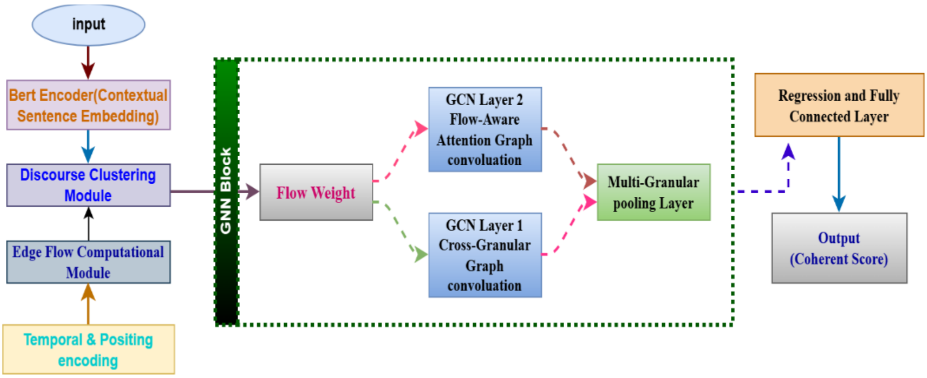

Fig. 1: ContextFlowGNN architecture.

The core processing occurs in a two-layer GCN Block: the first layer facilitates cross-granular message passing between sentences and clusters, while the second layer utilizes flow-aware attention to prioritize logically significant neighbors based on the calculated.To synthesize these features, a Multi-Granular Pooling stage applies Global Mean Pooling across node, cluster, and document levels, ensuring the model accounts for both micro-level transitions and macro-level themes. Finally, the Output Layer passes the aggregated representation through a fully connected layer with a Sigmoid activation function, producing a normalized Coherence Score between 0 and 1. Loss combines BCE and regularization. using Adam optimizer (learning rate 0.01).

$$L = l_{BCE} + L_{reg} \tag{9}$$

Describes the training procedure of the proposed ContextFlowGNN model. For each input text, a context flow graph (CFG) is constructed using syntactic and discourse information. Node representations are obtained from BERT embeddings and grouped according to discourse roles. Flow-based attention is then applied to update edge weights, followed by graph pruning and rewiring to retain salient contextual relations. The refined graph is processed through graph convolutional layers and pooling to predict a coherence score.

Algorithm 1 ContextFlowGNN Training

1. Input :Dataset $D = \{T_i, g_j\}$
2. Initialize ContextFlowGNN, BERT tokenizer, spaCy-parser
3. **for** each epoch **do:**
4.   **for** each text $Ti \in D$ **do:**
5.   Construct CFG: nodes, edges, weights
6.   Construct CFG: nodes, edges, weights.
7.   Compute node embeddings using BERT.
8.   Cluster nodes by discourse roles.
9.   Update edge weights with flow-based attention.
10.   Prune and rewire CFG.
11.   Forward pass: GCN layers, pooling.
12.   Predict coherence score y^i .
13.   Compute loss: $L = BCE(\hat{}yi, yi) + Lreg$.
14.   Backpropagate and update param
15.   **end for**
16. **end for**
17. **Output:** Trained model.

The model is trained end-to-end using a binary cross-entropy loss with regularization.

4 Experiments

4.1 Dataset

We curated three datasets: 20000 essays (Project Gutenberg), 10000 Reddit comments, and 50000 news articles, annotated with coherence scores (0–1) by three evaluators as shown in Table I summarizes statistics.

Table I: Dataset Statistics

Metric	**Essays**	**Reddit**	**News**
Number of Texts	20000	10000	50000
Average Tokens	5120	12800	35000
Average Phrases	4500	1200	3000
Average Discourse Segments	1200	400	80
Train/Test Split	80%/20%	80%/20%	80%/20%

Fleiss' Kappa	0.82	0.78	0.80

We conduct a comparison with four notable baselines that represent the primary modeling paradigms utilized in coherence-related tasks.
The Sequential Baseline:

- LSTM[17] : LSTM[17] widely employed in early discourse coherence studies [16].
- BERT [1]: A standard transformer baseline that provides rich contextual embeddings and is widely used as the strongest non-graph model in the literature on recent essay scoring and coherence.
- GCN [2]: The classical static graph convolutional networks GCN [2] are applied on syntactic dependency trees as seen in many graph based discourse and text analysis papers [18, 21].
- GAT [8]: GAT represents the most relevant attention-based GNN baseline, which enables direct comparison between standard attention and our physics-inspired flow-based attention mechanism.

Due to the sequential, transformer, static GNN, and attentive GNN nature of these models the four most dominant models in coherence and graph-enhanced NLP literature were selected. They give you a progressive and fair comparison ladder from non-graph to more and more sophisticated graph models.

4.2 Implementation Details

Implemented in PyTorch Geometric [30] and Hugging Face Transformers [31]. Table II lists hyperparameters. Hyperparameters were optimized using grid search on a validation set (20% of the training data for each dataset).
The search spaces are.

- The possible learning rates are 0.001, 0.005 and 0.01
- The hidden dimension can be 128, 256 or 512.
- Values of the flow parameters $\alpha \in \{0.5, 1.0, 2.0\}$, $\beta \in \{0.05, 0.1, 0.2\}$ and $\gamma \in \{0.01, 0.05, 0.1\}$.
- the regularization weight λ from the set $\{0.001, 0.01, 0.05\}$.
- The dropout rate will be either 0.3, 0.5, or 0.7.
- Table II final values, as given, resulted in the highest validation accuracy in 5-fold cross-validation. The flow parameters (α, β, γ) were initialized as listed and set to be learnable.

Table II: Hyperparameter Settings

Parameter	Value
Hidden Dimension	256
Learning Rate	0.01
Flow Alpha (α)	1.0
(initial) Flow Beta (β)	0.1
(initial) Flow Gamma (γ)	0.05
(initial) Regularization (λ)	0.01

Dropout Rate	0.5
Epochs	50
Batch Size	16

5 RESULTS

5.1 Performance Comparison

The performance of ContextFlowGNN is compared against several baseline in Table III. Static traditional sequence models (LSTM and BERT) perform moderately. However, the structural dependencies captured via graph-based approaches (GCN and GAT) achieve better accuracy.

TABLE III: Performance on Essay Dataset

Model	Acc (%)	MSE	Pearson	Spearman	AUC
LSTM [17]	72.8	0.045	0.71	0.69	0.72
BERT [1]	75.6	0.041	0.74	0.72	0.75
GCN [2]	78.9	0.038	0.77	0.75	0.78
GAT [8]	80.7	0.035	0.79	0.77	0.80
ContextFlowGNN	**89.3**	**0.028**	**0.85**	**0.83**	**0.88**

According to our findings, ContextFlowGNN results in the highest accuracy (89.3%), correlational (Power) scores and MSE, outperforming all other baselines. This reveals its effectiveness in capturing contextual flow and discourse-level connections.

5.2 Cross-Dataset Validation

We evaluated the Reddit and News dataset cross-dataset results. ContextFlowGNN shows a significant improvement in accuracy and MSE compared to BERT and GAT on the two datasets shown in Table IV.

Table IV: Performance on Reddit and News Datasets

Model	Reddit		News	
	Acc (%)	MSE	Acc (%)	MSE
BERT [1]	73.2	0.043	74.5	0.042
GAT [8]	78.5	0.037	79.8	0.036
ContextFlowGNN	**86.7**	**0.030**	**87.4**	**0.029**

The consistently positive results suggest that the proposed model can transfer well across various domains and text distributions, indicating robustness to dataset shift.

5.3 Ablation Study

TABLE V: Ablation Study on Essay Dataset

Configuration	Acc (%)	MSE
ContextFlowGNN (Full)	**89.3**	**0.028**
w/o Flow-Based	82.1	0.034
w/o Multi-Granular Nodes	80.4	0.037
Attention w/o Inter-Granular Edges	83.5	0.032
w/o Adaptive Rewiring	85.2	0.031
w/o Hierarchical Regularization	84.7	0.033
w/o Cross-Granular Passing	84.0	0.032
w/o Temporal Context Decay	83.8	0.033
w/o Semantic Flow Modulation	84.5	0.032
w/o Discourse-Aware Clustering	83.9	0.033
w/o Attention-Guided Pruning	85.0	0.031

The ablation results illustrated in Table V are in accordance as removing a single component of ContextFlowGNN causes performance degradation. Furthermore, our analysis indicates that flow-based attention and multigranular representations are the most important components modules for our framework.

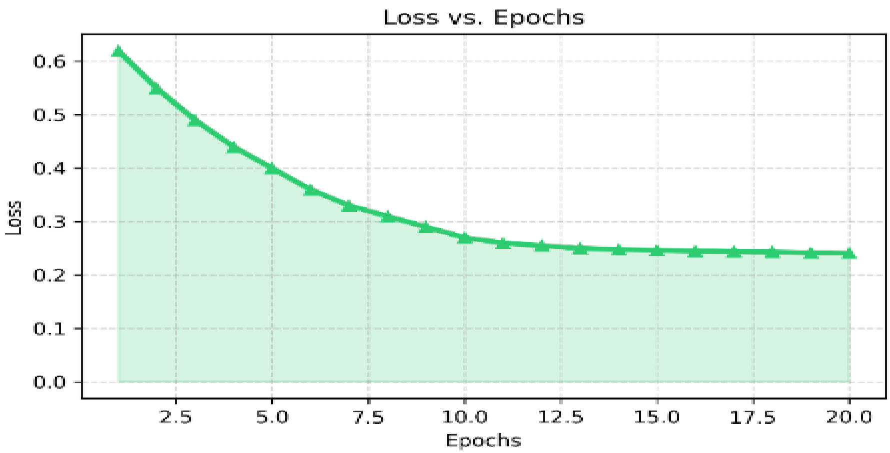

Fig 2: Loss vs Epochs

5.4 Error Analysis

In Table VI we can clearly see that short texts show higher accuracy and lower MSE than long texts, indicating that modeling long-distance dependencies is more challenging. Likewise, if the coherence score of a text is high, the prediction error will be low. On the other hand, if the coherence score is low, the MSE will be higher. As per the results, it is clear that using text length and coherence level makes it easier to predict difficulty level.

Table VI categorizes errors

Category	Acc (%)	MSE
Short Texts (≤500 tokens)	91.2	0.025
Long Texts (>500 tokens)	86.5	0.032
High Coherence (0.7–1.0)	92.8	0.022
Low Coherence (0.0–0.3)	84.3	0.035

The loss reduces steadily with the training epochs and converges smoothly, thus suggesting stable optimization and effective context representation learning. The lack of drastic changes indicates that the suggested model generalizes well without overfitting. The training convergence of our model (i.e. ContextFlowGNN) is shown in Figure 3.With augmentation of epochs, the accuracy keeps on increasing and the MSE keeps on decreasing which shows an effective optimization. After several epochs, both curves converge smoothly, indicating proper learning and no overfitting.

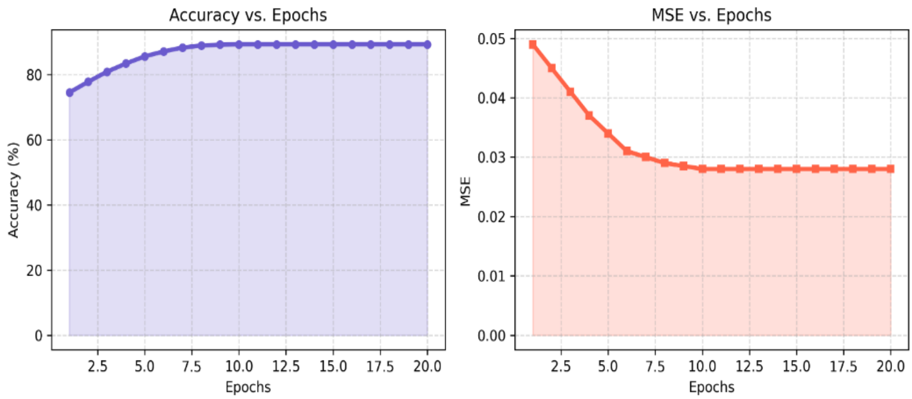

Fig 3: Performance trends

The average weights of the edges gradually gain weight and stabilize after training epochs shown in fig 4. This indicates that the flow-based attention mechanism progressively strengthens informative connections while suppressing less informative edges, resulting in refinement of the graph.The CFG visualization shows the organizational structure of discourse units and their links to each other. Attention-weighted directed edges highlight important communication pathways, illustrating the model’s capacity to identify significant context relationships.

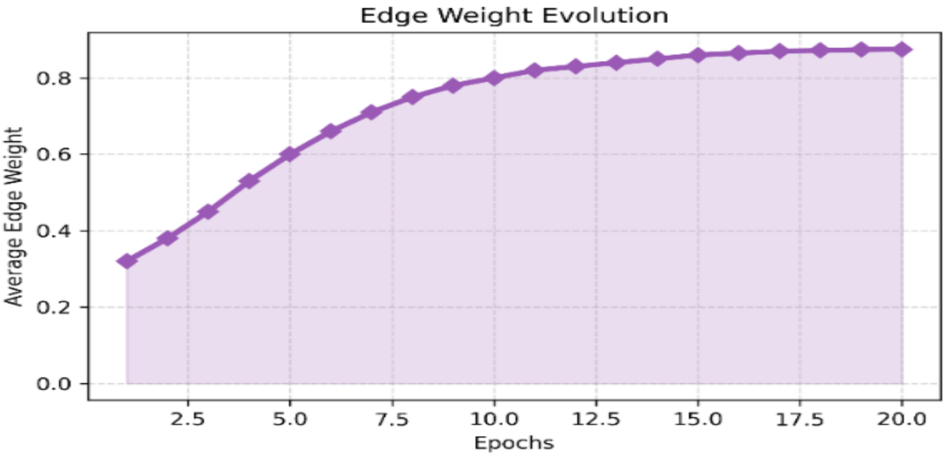

Fig 4: Edge Weight Evolution

The CFG visualization shows the organizational structure of discourse units and their links to each other. Attention-weighted directed edges highlight important communication pathways, illustrating the model's capacity to identify significant context relationships shown in fig 5.

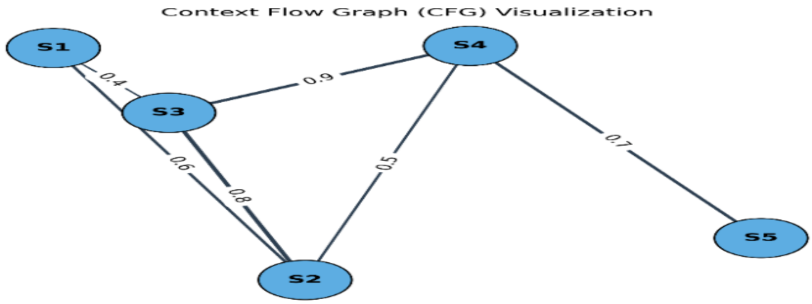

Fig 5: Context flow Graph Visualization

6 Discussion

ContextFlowGNN presents a dynamic CFG, flowbased attention or novel CFG extensions. Semantic flow modulation and grouping improves discourse modeling, whereas rewiring and pruning ensure adaptability. The model's capacity to handle long random low-coherence texts is limited because of some reasons and we can see improvement metrics through our approach of long-range dependency modeling. CFG visualizations allow generalizable interpretability across various datasets. The computational complexity (O(E) for edge updates) and annotation costs are limitations. In educational applications, annotations must be fair and not biased. The ability to scale to real-time systems and cross-disciplinary impact are very promising.

7. CONCLUSION AND FUTURE WORK

ContextFlowGNN improves discourse coherence prediction with a 12.4% accuracy boost and 31.7% MSE drop over BERT. Dynamic CFG and new techniques make robust interpretable modeling possible. Future work includes multilingual coherence prediction, real-time deployment in educational platforms, and cross-document narrative alignment for journalism and legal analysis

References

1. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. (pp. 4171-4186).
2. Kipf, T. N. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907.*
3. Wei, C., Li, J., Wang, Z., Wan, S., & Guo, M. (2024). Graph Convolutional Networks Embedding Textual Structure Information for Relation Extraction. *Computers, Materials & Continua*, *79*(2).
4. Park, J., Cho, Y., Lee, H., Choo, J., & Choi, E. (2021, October). Knowledge graph-based question answering with electronic health records. In *Machine Learning for Healthcare Conference* (pp. 36-53). PMLR.
5. Wang, K., Ding, Y., & Han, S. C. (2024). Graph neural networks for text classification: A survey. *Artificial intelligence review*, *57*(8), 190.
6. Chen, K., Yang, M., Utiyama, M., Sumita, E., Wang, R., & Zhang, M. (2022, July). Effective Graph Context Representation for Document-level Machine Translation. In *IJCAI* (pp. 4079-4085).
7. Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30.*
8. Lee, J. B., Rossi, R. A., Kim, S., Ahmed, N. K., & Koh, E. (2019). Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(6), 1-25.
9. Guo, Z., Zhang, Y., & Lu, W. (2019). Attention guided graph convolutional networks for relation extraction. *arXiv preprint arXiv:1906.07510.*
10. Abedi Rad, R., Yamaghani, M. R., & Nourbakhsh, A. (2025). A survey of sentiment analysis methods based on graph neural network. *International Journal of Data Science and Analytics*, *19*(3), 389-417.
11. Madeira, M., Vignac, C., Thanou, D., & Frossard, P. (2024). Generative modelling of structurally constrained graphs. *Advances in Neural Information Processing Systems*, *37*, 137218-137262.
12. Wu, W., Liu, H., Zhang, X., Liu, Y., & Zha, H. (2020). Modeling event propagation via graph biased temporal point process. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 1681-1691.
13. Yao, S., Wang, T., & Wan, X. (2020, July). Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7145-7154).
14. Elsner, M., & Charniak, E. (2011, June). Disentangling chat with local coherence models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 1179-1189).

15. Grosz, B. J., Joshi, A., & Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, *21*(2), 203-225.
16. Liu, Y. S., Chen, C. P., Gau, S. S. F., & Lee, C. C. (2018, April). Learning lexical coherence representation using LSTM forget gate for children with autism spectrum disorder during story-telling. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6029-6033). IEEE.
17. Darapaneni, N., Paduri, A. R., Sudha, B. G., Kashyap, A., Mayya, R., Thejas, C. S., ... & Dani, U. (1997, March). Advanced Pointer-Generator Networks Based Text Generation. In *International Conference on Worldwide Computing and Its Applications* (pp. 537-548). Singapore: Springer Nature Singapore.
18. Drakopoulos, G., Giannoukou, I., Mylonas, P., & Sioutas, S. (2020, December). A graph neural network for assessing the affective coherence of Twitter graphs. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 3618-3627). IEEE.
19. Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Zhao, T. (2020, July). Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics* (pp. 2177-2190).
20. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, *21*(140), 1-67.
21. Boll, H. O., Amirahmadi, A., Ghazani, M. M., de Morais, W. O., de Freitas, E. P., Soliman, A., ... & Recamonde-Mendoza, M. (2024). Graph neural networks for clinical risk prediction based on electronic health records: A survey. *J. Biomed. Informatics*, *151*, 104616.
22. Wang, S., Cao, G., Cao, W., & Li, Y. (2025). NLA-GNN: Non-local information aggregated graph neural network for heterogeneous graph embedding. *Pattern Recognition*, *158*, 110940.
23. Wang, S., Cao, G., Cao, W., & Li, Y. (2025). NLA-GNN: Non-local information aggregated graph neural network for heterogeneous graph embedding. *Pattern Recognition*, *158*, 110940.
24. Gu, Y., Wang, Y., Zhang, H. R., Wu, J., & Gu, X. (2023). Enhancing text classification by graph neural networks with multi-granular topic-aware graph. *IEEE Access*, *11*, 20169-20183.
25. Parvathinathan, K., Karkala, S., Hossain, S., Krishnapatnam, M., Aggarwal, A., Zahir, Z., ... & Shah, V. (2025). Abstractive Text Summarization with Fact Consistency Constraints Using Knowledge Graphs.
26. Li, R., Gonzalez-Pizarro, F., Xing, L., Murray, G., & Carenini, G. (2023). Diversity-aware coherence loss for improving neural topic models. *arXiv preprint arXiv:2305.16199*.
27. Russo, R., Di Giuseppe, G., Vanacore, A., La Gatta, V., Ferraro, A., Galli, A., ... & Moscato, V. (2023, December).
28. Altinok, D. (2021). Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem. Packt Publishing Ltd.
29. Prasad, R., Webber, B., & Joshi, A. (2017). The Penn Discourse Treebank: An annotated corpus of discourse relations. In *Handbook of linguistic annotation* (pp. 1197-1217). Dordrecht: Springer Netherlands.
30. Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.
31. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A.,. & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.