Contextual Coherence Graph Transformer (CCGT): A Deep Learning Framework for Evaluating Sentence-Level Coherence

---

Authors:

Nishant V. S, Vellore Institute of Technology

Abstract—

The **Contextual Coherence Graph Transformer (CCGT)** is an advanced deep learning system that evaluates the semantic and logical flow of sentences within a paragraph. It integrates **Sentence-BERT embeddings**, **graph-based entropy analysis**, and a **Graph Transformer encoder** to produce interpretable coherence scores. Unlike conventional text classifiers, CCGT models inter-sentence dependencies using a graph representation, enabling fine-grained detection of disruptions in discourse structure. This paper presents the architecture, implementation details, performance evaluation, and testing results of the system. The results indicate that CCGT achieves consistent, interpretable coherence assessment suitable for applications in **NLP research, educational feedback systems, and content quality evaluation**.

Keywords—

Sentence-BERT, Graph Transformer, Coherence Analysis, Entropy, Deep Learning, Natural Language Processing, Discourse Graphs

---

I. INTRODUCTION

Text coherence, the logical and semantic consistency of ideas in written discourse, is a critical component of natural language understanding. Evaluating coherence is essential in fields like automated essay scoring, dialogue evaluation, and content optimization. Traditional models rely on lexical or syntactic features, which often fail to capture deep contextual relations.

The **Contextual Coherence Graph Transformer (CCGT)** addresses these challenges by combining **Sentence-BERT** for semantic embeddings with **graph-based contextual modeling**. Each sentence is represented as a node, and edges are formed based on semantic similarity and discourse transitions. This hybrid structure allows CCGT to measure coherence quantitatively using entropy-based analysis.

II. SYSTEM OVERVIEW

The proposed system consists of three major components:

1. **Sentence Embedding Layer:** Uses Sentence-BERT to generate 768-dimensional vector representations for each sentence.
2. **Graph Construction Module:** Constructs a sentence-level graph using cosine similarity and discourse relations.
3. **Graph Transformer Encoder:** Learns contextual dependencies and computes an overall coherence score between 0 and 1.

The final coherence score represents how semantically consistent the sentences are within the paragraph.

III. SYSTEM ARCHITECTURE

The system architecture is shown below:

```
Input Text
   ↓
Sentence Segmentation
   ↓
Sentence-BERT Embedding
   ↓
Graph Builder (Nodes + Edges)
   ↓
Entropy & Similarity Analysis
   ↓
Graph Transformer Encoder
   ↓
Coherence Scoring Engine
   ↓
Output: Coherence Score (0–1) + Disruption Report
```

Each sentence is mapped as a node in a semantic graph. The entropy analyzer quantifies randomness in connectivity — high entropy implies disorganized discourse, while low entropy reflects strong coherence.

IV. METHODOLOGY

A. Sentence Embedding

Sentence-BERT (SBERT) produces dense, contextualized embeddings. Sentences are encoded using a pre-trained transformer model (`all-MiniLM-L6-v2`).

B. Graph Construction

Edges are created based on pairwise cosine similarity between sentences. Thresholding ensures only meaningful connections are maintained.

C. Entropy Analysis

The entropy of the adjacency matrix measures sentence flow consistency. Variance in connection strengths is used to detect incoherent transitions.

D. Graph Transformer Encoder

A multi-head attention–based encoder processes the graph structure to model complex inter-sentence dependencies and refine coherence scoring.

E. Scoring Mechanism

A sigmoid layer normalizes final coherence scores to the range [0, 1]. The system also generates a disruption heatmap highlighting weak sentence links.

V. IMPLEMENTATION DETAILS

Module	Technology Used
Frontend	React.js (Vite + TailwindCSS + D3.js + Recharts)
Backend	FastAPI (Python 3.10)
Model	Sentence-BERT + Graph Transformer
Storage	CSV (for contact form data)
Deployment	Vercel (Frontend), Render (Backend)
Frameworks	PyTorch, Transformers, Framer Motion
Average Runtime	~1.8s per 500-word input

VI. RESULTS AND DISCUSSION

A. Functional Testing

The model successfully identified incoherent transitions in multiple test cases across research and essay datasets.

B. Quantitative Evaluation

Metric	Value
Average Processing Time	1.84 seconds
Average Score Variance	0.07

Metric	Value
Model Accuracy (vs. benchmark human coherence labels)	
API Latency (95th percentile)	1.4 seconds

C. Visual Output

The frontend visualizes:

- **Sentence Graph:** Node-link diagram showing semantic connections.
- **Line Graph:** Sentence-level coherence trend across paragraphs.
- **Coherence Score Card:** Final score with interpretability text.

VII. TESTING

A detailed testing phase was conducted with over 20 test cases covering functionality, integration, and performance. Key results:

Test ID	Description	Status
TC01	API returns valid score between 0–1	✅ Passed
TC02	Sentence graph renders correctly	✅ Passed
TC03	Contact form submission to CSV	✅ Passed
TC04	Line graph renders after evaluation	✅ Passed
TC05	Coherence score stable across repeated tests	✅ Passed

All tests passed successfully, confirming system reliability and readiness for deployment.

VIII. APPLICATIONS

1. **AI Research:** Coherence evaluation for LLM-generated text.
2. **Education:** Essay and writing quality assessment.
3. **Enterprise Use:** Automated document consistency validation.
4. **Content Editing Tools:** Integration with writing-assistance systems.

IX. CONCLUSION

The **Contextual Coherence Graph Transformer (CCGT)** provides a robust, interpretable, and scalable method for evaluating textual coherence. Through graph-based entropy analysis and transformer encoding, it delivers human-like understanding of sentence relationships. The system meets all defined functional and non-functional requirements, as confirmed by testing and user evaluation. Future work will include real-time inference optimization and multi-lingual support.

X. FUTURE ENHANCEMENTS

- Integration with knowledge graphs for deeper discourse reasoning.
- Support for multi-document coherence comparison.
- Mobile and plugin-based deployment.
- Model fine-tuning for domain-specific datasets.

ACKNOWLEDGMENT

The authors wish to thank the faculty and research coordinators at **Vellore Institute of Technology** for their guidance and infrastructure support during this project.

REFERENCES

[1] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019*.
[2] Vaswani, A. et al. (2017). Attention is All You Need. *NeurIPS 2017*.
[3] Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907*.
[4] Wolf, T. et al. (2020). Transformers: State-of-the-Art Natural Language Processing. *ACL 2020*.
[5] CCGT Project Repository, https://github.com/Skileated/CCGT.