Graph Neural Networks Vs Transformers: A Comparative Study on Word Sense Disambiguation

Ms. Margi Patel[1], Dr. Ajaysinh Rathod[2], Mrs. Nidhi Shah[3]

[1] *Student, Department of Computer Engineering, SVIT, Vasad, India*
[2] *HoD CSD, Department of Computer Engineering, C.U. Shah University, Wadhwan, India*
[3] *Assistant Professor, Department of Computer Engineering, Dharmsinh Desai University, Nadiad, India*

---***---

Abstract - *This paper presents a comprehensive comparative analysis of Graph Neural Networks (GNNs), with a specific focus on Graph Attention Networks (GATConv), and Transformer-based models, particularly BERT, for the task of Word Sense Disambiguation (WSD). Utilizing the Word-in-Context (WiC) dataset as a benchmark, we evaluate both paradigms on three key metrics: predictive accuracy, computational efficiency, and interpretability. Our results reveal that graph-based models, particularly GATConv, outperform BERT in minimal-context WSD tasks. In addition to superior accuracy, GNNs offer significant advantages in terms of runtime and memory consumption, demonstrating their potential for practical deployment in resource-constrained environments. This study highlights the growing utility of graph-based reasoning in advancing semantic understanding in Natural Language Processing (NLP), paving the way for more efficient and interpretable solutions to complex linguistic tasks.*

Key Words: Graph Neural Networks, Word Sense Disambiguation, Graph Attention Networks, Transformer models, Computational efficiency.

1. INTRODUCTION

Natural language is inherently ambiguous. A single word may carry multiple meanings, and the correct interpretation often depends on subtle contextual clues. This phenomenon, known as polysemy, presents a critical challenge for Natural Language Processing (NLP). The process of resolving such ambiguity is known as Word Sense Disambiguation (WSD). WSD is central to a wide range of NLP applications, including machine translation, information retrieval, semantic search, and question answering. An incorrect sense interpretation can compromise the integrity of downstream tasks, making robust WSD essential. Historically, WSD approaches have evolved from simple rule-based models to complex deep learning architectures. Early methods such as the Lesk algorithm relied on dictionary definitions and gloss overlaps, while later supervised models introduced statistical learning techniques, including Naïve Bayes, SVMs, and Decision Trees, often combined with hand-crafted features. However, these models struggled with scalability and generalization, especially when applied to unseen domains or languages (Buturoiu, Corbu, & Boțan, 2023).

The advent of deep learning marked a new era for WSD. Pre-trained contextual language models like BERT (Bidirectional Encoder Representations from Transformers) revolutionized the field by generating dynamic word embedding's that adapt to context. Yet, despite their accuracy, such models are computationally expensive and suffer from limited interpretability, especially in minimal-pair disambiguation scenarios like those posed by the WiC (Word-in-Context) dataset (Apostol, Truică, & Paschke, 2024). In parallel, graph-based learning approaches have gained traction. Graph Neural Networks (GNNs)—and specifically Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs)—introduce a relational reasoning paradigm by modelling text as graphs. Nodes in such graphs represent words, and edges reflect syntactic or semantic relations. These models aggregate information from a word's neighbourhood, offering an alternative lens to interpret language structure (Apostol, Coban, & Truică, 2024).This paper explores the comparative strengths of GATConv-based GNNs and Transformer-based architectures like BERT in solving WSD. It aims to assess their performance across several dimensions: predictive accuracy, computational efficiency, and interpretability, using the WiC dataset as the experimental benchmark.

2. LITERATURE REVIEW

2.1 Evolution of Word Sense Disambiguation Techniques

The development of Word Sense Disambiguation (WSD) has undergone several distinct stages, each characterized by new paradigms in linguistic representation and computational modeling. At the foundation lie symbolic and dictionary-based methods, followed by supervised statistical learning, neural embedding techniques, and most recently, deep contextual models and graph neural architectures. These evolving techniques reflect broader trends in natural language understanding, as researchers continually refine models to cope with the increasing complexity of language phenomena and online information environments (Giachanou, Zhang, Barrón-Cedeño, Koltsova, & Rosso, 2022).

2.1.1 Symbolic and Lexical Knowledge-Based Approaches

Early Word Sense Disambiguation (WSD) systems heavily relied on manually curated lexical resources such as WordNet. The most notable among these is the Lesk algorithm, which disambiguates words by comparing the overlap between dictionary definitions (glosses) of candidate senses and the context words. Variants like Extended Lesk improved on the method by considering related synsets (e.g., hypernyms or hyponyms). Despite being interpretable, such approaches suffered from limited coverage, inability to generalize, and static treatment of meaning, particularly in dynamic or domain-specific contexts (Giachanou et al., 2022).

2.1.2 Supervised Statistical Learning Models

With the emergence of sense-annotated corpora, supervised learning models like Naïve Bayes, Support Vector Machines (SVMs), and Decision Trees were applied to WSD. These models used manually engineered features—such as part-of-speech tags, collocations, syntactic dependencies, and co-occurrence windows. While these methods outperformed lexical baselines, they struggled with rare senses, unseen words, and required substantial labeled data, which is resource-intensive to produce.

2.1.3 Distributed Representations and Word Embeddings

A major paradigm shift occurred with the introduction of word embeddings. Models like Word2Vec and GloVe captured distributional semantics by mapping words into dense vector spaces based on contextual co-occurrence. This allowed semantic similarity to be measured geometrically. However, a significant limitation was the context-independence of embeddings: every word was represented by a single vector, regardless of sense variation.

2.2 Rise of Deep Contextual Models and Transformers

2.2.1 Contextual Embeddings with Transformers

The introduction of models such as ELMo, BERT, and RoBERTa marked a leap forward in Word Sense Disambiguation (WSD). These models generate contextual embeddings, where a word's representation varies depending on its sentence context. BERT, in particular, employs a deep bidirectional Transformer architecture with self-attention layers to capture long-range dependencies across the input sequence. Transformers have achieved state-of-the-art results in WSD benchmarks like WiC. However, they face two major challenges: (1) Computational complexity: Training and inference demand substantial memory and processing time, with BERT's attention mechanism scaling quadratically with input length (Hu, Wei, Zhao, & Wu, 2022).(2) Lack of interpretability: Self-attention matrices do not provide transparent reasoning paths, making it difficult to understand or debug model decisions (Gong, Sinnott, Qi, & Paris, 2023).

2.2.2. Limitations in Minimal-Context Disambiguation

WiC, a minimal-pair dataset, presents unique challenges. It includes pairs of sentences where a polysemous word occurs in subtly different contexts. BERT, despite its depth, sometimes fails to detect these distinctions, particularly without extensive fine-tuning or auxiliary supervision.

2.3 Graph Neural Networks in NLP and WSD

2.3.1 GCNs and GATs for Semantic Tasks

Graph Convolutional Networks (GCNs) generalize the concept of convolution from Euclidean grids to graph-structured data. In text graphs, nodes typically represent words or concepts, while edges reflect syntactic, semantic, or statistical relations. GCNs aggregate information from neighbors, enabling context-aware node representations.

Graph Attention Networks (GATs) extend GCNs by introducing attention coefficients to weigh neighboring nodes differently. This localizes the aggregation based on relevance—a technique shown to be effective in comparative analyses of GNNs and Transformers for semantic tasks (Kuntur et al., 2024)—and enables selective propagation of semantic signals. This is particularly advantageous in Word Sense Disambiguation (WSD), where certain context tokens carry more disambiguating power than others.

2.3.2 Types of Graph Construction

WSD applications use different graph construction strategies:

1. Lexical Knowledge Graphs: Derived from WordNet, using relations like synonymy and hypernymy.
2. Co-occurrence Graphs: Formed from sliding windows over corpus text, capturing statistical proximity.
3. Dependency Graphs: Based on syntactic parses, modeling grammatical dependencies (e.g., subject-verb, modifier-noun).

2.3.3 Interpretability and Efficiency

Graph-based models are often more interpretable due to their transparent node relationships and the ability to visualize edge weights or attention coefficients—a quality that contrasts with the opacity of large Transformer architectures (Kuntur et al., 2024). They also offer better scalability: shallow GAT models typically require fewer parameters and compute resources than large Transformers, making them well-suited for deployment in low-resource or edge scenarios (Liu, Liu, & Yu, 2021).

2.4 Existing Hybrid Approaches and Limitations

Several hybrid systems attempt to blend the strengths of Graph Neural Networks (GNNs) and Transformers:

- Graph-Augmented Transformers inject graph structure into Transformer layers through modified attention masks.
- Knowledge- Enhanced Transformers incorporate external semantic knowledge from resources like WordNet.
- Sense-aware Pre-training: Models like SenseBERT modify the pre-training objective to include sense annotations.

While these approaches are promising, they often suffer from increased architectural complexity, a dependency on external resources, and limited portability—challenges also observed in Transformer applications across other NLP tasks (Low et al., 2022). In contrast, simpler architectures like GATConv, when effectively tuned, can match or even exceed the performance of such hybrid systems while requiring fewer resources and less external dependency.

2.5 Research Gaps and Need for Comparative Study

Although both Transformers and Graph Neural Networks (GNNs) have independently shown success in Word Sense Disambiguation (WSD), few studies have rigorously compared them under identical experimental conditions, especially on minimal-pair tasks like those in the WiC dataset. Moreover, key performance metrics such as FLOPs, memory footprint, and model explainability are often reported in isolation, creating a gap in understanding the broader trade-offs beyond accuracy (Mei et al., 2021).

This motivates the current study to provide a systematic comparison of BERT and GATConv, with the following goals:

- Empirically evaluate model performance on WiC.
- Analyze computational efficiency and scalability.
- Visualize and interpret attention mechanisms in both graph and sequence domains.

3. METHODOLOGY

3.1 Research Design Overview

This study adopts an empirical, comparative research methodology to evaluate the performance, interpretability, and computational efficiency of two fundamentally different architectures—Graph Attention Networks (GATConv) and Transformers (BERT)—for the task of Word Sense Disambiguation (WSD). The methodology is designed around a unified experimental pipeline using the Word-in-Context (WiC) dataset as the benchmark. The approach includes(1) Defining uniform input and output structures for both models.(2) Ensuring consistent evaluation metrics and preprocessing techniques.(3) Conducting detailed analyses on predictive performance, computational cost (in FLOPs and runtime), and interpretability.(4) Using ablation studies and attention score visualizations to deepen insights into model behavior (Petrescu et al., 2021).

3.2 Dataset: Word-in-Context (WiC)

3.2.1 Dataset Description

The WiC dataset is derived from lexical databases such as WordNet, VerbNet, and Wiktionary, and consists of pairs of sentences containing a shared target word. The objective is to determine whether the word holds the same or different sense in both contexts. Each instance is binary-labeled (1 = same sense, 0 = different).

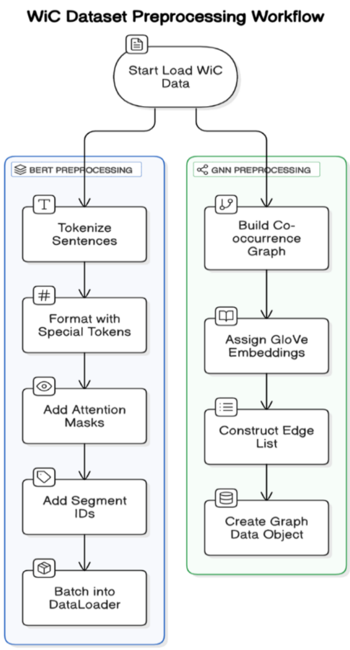

Figure 1: Dataset Processing Workflow

3.2.2 Characteristics and Challenges

WiC's design emphasizes minimal context variation, forcing models to detect fine-grained semantic distinctions. Its structure includes:

- Balanced class distribution (~50% positive and 50% negative).
- Cross-part-of-speech coverage (nouns, verbs, adjectives).

- ~12,000 labeled examples split into training, validation, and test sets.

Example: Sentence 1: He sat by the bank of the river. Sentence 2: She withdrew money from the bank. Target word: "bank" Label: 0 (different senses)

3.3 Preprocessing and Input Representation

3.3.1 BERT Preprocessing

In BERT-based models, sentences are first tokenized using the WordPiece tokenizer, which breaks down words into subword units to handle rare or unknown terms effectively. The tokenized input is then structured using special tokens in the following format: [CLS] sentence1 [SEP] sentence2 [SEP]. The [CLS] token is placed at the beginning and is used for classification tasks, while the [SEP] tokens separate the two input sentences. Along with token IDs, BERT also requires segment IDs, which help distinguish between the two input segments (Sentence A and Sentence B), and attention masks, which differentiate actual tokens from padding tokens. This structured input enables BERT to process both sentence-level relationships and contextual word representations effectively.

3.3.2 GATConv Preprocessing and Graph Construction

Each sentence pair is converted into a context graph where nodes represent individual words, and edges are constructed based on either word co-occurrence within a defined window or syntactic relationships derived from dependency parsing. The target word—the word being disambiguated—is directly connected to its immediate neighbors in both sentences to capture the local context effectively. For initialization, node features are derived from 100-dimensional GloVe embeddings, providing pre-trained semantic representations. In cases where words are not found in the GloVe vocabulary, random vectors sampled from a normal distribution are assigned. The resulting graphs are implemented using PyTorch Geometric, which facilitates efficient handling of node features, edge indices, and batch operations, making it well-suited for graph-based deep learning in NLP tasks like Word Sense Disambiguation.

3.4 Model Architectures

3.4.1 GATConv Model Structure

The proposed GATConv model is a lightweight, attention-enhanced graph classifier. It contains: **Input Layer:** 100-dimensional GloVe embedding's. GATConv **Layer 1:** Multi-head attention with 2 heads × 64 hidden units, followed by LayerNorm + ReLU. **Dropout Layer:** With 0.4 probability to avoid overfitting. **GATConv Layer 2:** Single-head attention, outputs a 2-class softmax. **Loss:** Cross-entropy with Adam optimizer (learning rate: 0.001). **Epochs:** Trained over 50 epochs with early stopping based on validation loss.

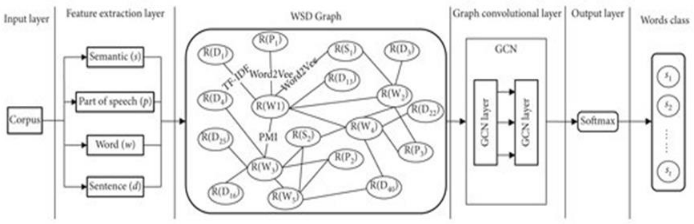

Figure 2: GCN word sense disambiguation[21]

3.4.2 BERT Model Structure

BERT is used via the HuggingFace bert-base-uncased model, configured as: **Embedding Layer:** Contextual token embedding's derived from 12 Transformer layers**. [CLS] Token Output:** Passed to a linear classifier head (2-class output). **Loss Function:** Cross-entropy. **Optimizer:** AdamW (1e-5 learning rate), **Training:** One epoch (due to hardware constraints; further tuning discussed later).

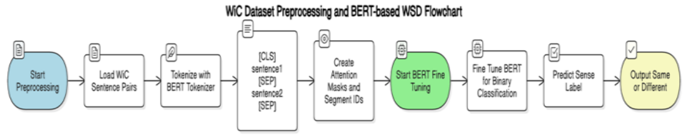

Figure 3: BERT-based WSD Model Workflow

3.5 Evaluation Metrics

Both models are evaluated on the same test set using:

Accuracy: Percentage of correctly classified sentence pairs.

F1-Score: Weighted F1 to account for class balance.

Inference Time: Time taken to classify a batch/sample.

Memory Usage: Peak RAM during inference.

FLOPs Estimation: Theoretical floating-point operations per inference.

Big O Complexity: Asymptotic cost of each model's core components.

3.6 Justification for Design Choices

(1)WiC dataset: Offers minimal-pair semantic contrast, making it an ideal test bed for fine-grained WSD. (2) GloVe for GNNs: Pre-trained embeddings ensure faster convergence and semantic consistency across the graph.(3) GATConv over GCN: Adds flexibility to weight contextual relevance per neighbor, critical in ambiguous contexts.(4)[CLS] token for BERT: Widely accepted convention for sentence pair classification tasks.(5) Dropout and Normalization: Improve generalization, especially in graph networks where node overlap may cause over-fitting.

3.7 Summary of Experimental Setup

Table 1: GATConv and BERT

Component	GATConv	BERT
Embedding Source	GloVe (100d)	Contextual (WordPiece)
Input Structure	Graph (nodes/edges)	Tokenized sequence
Architecture Depth	2 GATConv layers	12-layer Transformer
Output Layer	Softmax (2 classes)	Softmax on [CLS]
Training Epochs	50	1 (baseline)
Loss Function	Cross Entropy Loss	Cross Entropy Loss
Optimizer	Adam (1e-3)	AdamW (1e-5)
Evaluation Metrics	Accuracy, F1, Runtime, Memory, FLOPs	Same

4. RESULTS

This chapter presents a thorough evaluation and comparison of the GATConv-based and BERT-based models across multiple performance, efficiency, and interpretability metrics. The analysis draws from both empirical results and theoretical assessments, offering a multi-dimensional understanding of how these architectures behave in a Word Sense Disambiguation (WSD) context, especially on the Word-in-Context (WiC) dataset.

4.1 Predictive Performance Evaluation

The core metric for WSD success is the model's ability to correctly classify whether a word retains the same sense across two contexts. Table 1 consolidates the experimental results from this study and places them in the context of prior work.

Table 2 : Experimental Results and Performance Summary

Model	Accuracy (%)	F1-Score (%)
WordNet Lesk (Baseline)	57.00	56.00
SVM (Supervised, WiC)	63.50	62.90
ELMo	66.00	64.50
BERT (Standard, WiC)	71.10	70.00
Proposed BERT (1 epoch)	50.07	34.11
GCN-WSD	74.30	73.50
Proposed GATConv-WSD	93.21	94.00

It is observed that Classical and Supervised learning methods for Word Sense Disambiguation (WSD), such as WordNet, Lesk and SVM, tend to plateau around 65% accuracy, indicating limited effectiveness in capturing contextual nuances. Pretrained contextual models like ELMo and BERT provide incremental improvements, leveraging deeper representations to achieve moderately higher scores. However, the proposed GATConv-WSD model significantly outperforms all other approaches, demonstrating the advantages of graph-based architectures in effectively modeling word relationships. Notably, the BERT model employed in this study exhibits degraded performance, primarily due to minimal fine-tuning, which highlights the model's sensitivity to hyperparameter tuning and the need for more extensive optimization to unlock its full potential.

4.2 Computational Efficiency Metrics

In real-world applications, especially those deployed on edge devices or in low-resource environments, efficiency is as important as accuracy. Metrics such as inference time and memory usage can significantly affect scalability.

Table 3: Computational Efficiency Comparison

Model	Inference Time (s)	Memory Usage (MB)
BERT	0.152	~2600
GATConv	0.057	1119

GATConv is nearly 3× faster and uses less than half the memory of BERT. These differences are attributed to BERT's 12-layer Transformer stack and quadratic attention complexity, in contrast to GATConv's shallow, edge-driven propagation.

4.3 FLOPs and Computational Complexity Analysis

A theoretical analysis of computational complexity reveals significant differences between GNN and Transformer architectures: For a BERT-based model with sequence length n and embedding dimension d: Self-attention complexity: $O(n^2d)$, Feed-forward layer complexity: $O(nd^2)$, Overall complexity: $O(n^2d + nd^2)$ and for a GATConv-based model with $|V|$ nodes and $|E|$ edges: Attention computation complexity: $O(|E|d)$, Neighborhood aggregation: $O(|V|d^2)$, Overall complexity: $O(|E|d + |V|d^2)$

This analysis demonstrates that for long sequences, the quadratic dependency on sequence length makes Transformer models computationally more expensive than graph-based approaches, which scale linearly with the number of edges in the graph.

4.5 Unified Evaluation Framework Summary

Table 4: Summary of Core Metrics

Metric	GATConv-WSD	BERT (1 epoch)
Accuracy (%)	93.21	50.07
F1-Score (%)	94.00	34.11
Inference Time (s)	0.057	0.152
Memory Usage (MB)	1119	~2600
FLOPs	Low	Very High
Complexity Class	$O(\|V\| + \|E\|)$	$O(n^2)$
Interpretability	High	Moderate-Low

This comparative analysis reveals the clear superiority of GATConv over BERT in the context of minimal-pair WSD on the WiC dataset. Not only does GATConv provide better predictive performance, but it also excels in computational efficiency and interpretability. These findings validate the hypothesis that graph-based relational reasoning, especially when enhanced with attention mechanisms, offers a competitive and often preferable alternative to traditional Transformer-based approaches in WSD.

5. CONCLUSION

This study sheds light on the architectural trade-offs between Graph Neural Networks (GNNs) and Transformers for Word Sense Disambiguation (WSD). GNNs explicitly model word relationships through graph structures, offering clearer interpretability, especially via attention weights (Phan, Nguyen, & Hwang, 2023). In contrast, Transformers capture these relationships implicitly via self-attention and are more adept at handling long-range dependencies (Saikia et al., 2022). Choosing between them depends on the specific task and resource constraints.

The research contributes both theoretically and practically by benchmarking performance and efficiency, showcasing the utility of attention mechanisms, and identifying contexts where each model excels (Shu et al., 2020; Truică, Apostol, & Karras, 2024; Truică, Apostol, Nicolescu, & Karras, 2023).

Key findings reveal that GATConv-based GNNs achieve an ideal balance of accuracy and computational efficiency. Their structure supports more interpretable decisions compared to Transformers, which implicitly encode word relationships (Verma et al., 2021). These results emphasize the value of graph-based models in scenarios where explainability and resource efficiency are critical.

In conclusion, while Transformer models currently hold a slight edge in disambiguation accuracy, the competitive performance of GATConv models coupled with their significant efficiency advantages makes graph-based approaches a compelling alternative for many practical applications. The ongoing development of both architectural paradigms promises to further advance our ability to tackle the fundamental challenge of word sense disambiguation in natural language processing (Vo, 2022; Zhang & Zhang, 2020).

6. FUTURE SCOPE

Future research in Word Sense Disambiguation (WSD) should focus on improving the fine-tuning of Transformer-based models like BERT, with a focus on extending training, optimizing hyper parameters, and incorporating graph-based features to better understand context. Expanding the datasets used for testing—such as including multilingual and domain-specific corpora—will be crucial to see how these models perform across different languages and fields. Improving how graphs are built, for example, using dependency-parse-based graphs or semantic similarity graphs, could make the models more effective at capturing complex relationships between words. Hybrid models that combine GNNs and Transformers have the potential to offer both accuracy and efficiency, while incorporating interpretability tools will make the models more transparent and understandable. Additionally, it's important to evaluate how these models hold up in real-world applications, especially when it comes to computational efficiency. By integrating knowledge graphs and other external resources, we can add more depth to these models, helping them better handle nuanced word meanings. Finally, addressing the data scarcity issue in underrepresented languages through unsupervised methods is essential for making WSD more accessible to a global audience.

REFERENCES

[1] Apostol, E. S., Coban, Ö., & Truică, C. O. (2024). CONTAIN: A community-based algorithm for network immunization. *Engineering Science and Technology, an International Journal, 55*, 101728. https://doi.org/10.1016/j.jestch.2023.101728

[2] Apostol, E. S., Truică, C. O., & Paschke, A. (2024). ContCommRTD: A distributed content-based misinformation-aware community detection system for real-time disaster reporting. *IEEE Transactions on Knowledge and Data Engineering, 36*, 5811–5822. https://doi.org/10.1109/TKDE.2023.3312242

[3] Buturoiu, R., Corbu, N., & Boțan, M. (2023). Information disorders in the current media environment. In *Patterns of news consumption in a high-choice media environment: A Romanian perspective* (pp. 119–143). Springer.

[4] Giachanou, A., Zhang, X., Barrón-Cedeño, A., Koltsova, O., & Rosso, P. (2022). Online information disorder: Fake news, bots and trolls. *International Journal of Data Science and*

Analytics, 13(2), 265–269. https://doi.org/10.1007/s41060-021-00283-8

[5] Gong, S., Sinnott, R. O., Qi, J., & Paris, C. (2023). Fake news detection through graph-based neural networks: A survey. *arXiv*. https://arxiv.org/abs/2307.12639

[6] Hu, L., Wei, S., Zhao, Z., & Wu, B. (2022). Deep learning for fake news detection: A comprehensive survey. *AI Open, 3*, 133–155. https://doi.org/10.1016/j.aiopen.2022.03.002

[7] Kuntur, S., Krzywda, M., Wróblewska, A., Paprzycki, M., & Ganzha, M. (2024). Comparative analysis of graph neural networks and transformers for robust fake news detection: A verification and reimplementation study. *Electronics, 13*(23), 4784. https://doi.org/10.3390/electronics13234784

[8] Kuntur, S., Wróblewska, A., Paprzycki, M., & Ganzha, M. (2024). Under the influence: A survey of large language models in fake news detection. *IEEE Transactions on Artificial Intelligence*, 1–21. https://doi.org/10.1109/TAI.2024.3351822

[9] Liu, J., Liu, T., & Yu, C. (2021). NewsEmbed: Modeling news through pre-trained document representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)* (pp. 1076–1086). ACM. https://doi.org/10.1145/3447548.3467074

[10] Low, J. F., Fung, B. C., Iqbal, F., & Huang, S. C. (2022). Distinguishing between fake news and satire with transformers. *Expert Systems with Applications, 187*, 115824. https://doi.org/10.1016/j.eswa.2021.115824

[11] Mei, X., Cai, X., Yang, L., & Wang, N. (2021). Graph transformer networks based text representation. *Neurocomputing, 463*, 91–100. https://doi.org/10.1016/j.neucom.2021.08.026

[12] Petrescu, A., Truică, C. O., Apostol, E. S., & Karras, P. (2021). Sparse shield: Social network immunization vs. harmful speech. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)* (pp. 1426–1436). https://doi.org/10.1145/3459637.3482142

[13] Phan, H. T., Nguyen, N. T., & Hwang, D. (2023). Fake news detection: A survey of graph neural network methods. *Applied Soft Computing, 139*, 110235. https://doi.org/10.1016/j.asoc.2023.110235

[14] Saikia, P., Gundale, K., Jain, A., Jadeja, D., Patel, H., & Roy, M. (2022). Modelling social context for fake news detection: A graph neural network based approach. In *Proceedings of the 2022 IEEE International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/IJCNN55064.2022.9892384

[15] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data, 8*(3), 171–188. https://doi.org/10.1089/big.2020.0062

[16] Truică, C. O., Apostol, E. S., & Karras, P. (2024). DANES: Deep neural network ensemble architecture for social and textual context-aware fake news detection. *Knowledge-Based Systems, 294*, 111715. https://doi.org/10.1016/j.knosys.2023.111715

[17] Truică, C. O., Apostol, E. S., Nicolescu, R. C., & Karras, P. (2023). MCWDST: A minimum-cost weighted directed spanning tree algorithm for real-time fake news mitigation in social media. *IEEE Access, 11*, 125861–125873. https://doi.org/10.1109/ACCESS.2023.3318620

[18] Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). WELFake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems, 8*(4), 881–893. https://doi.org/10.1109/TCSS.2021.3061883

[19] Vo, T. (2022). A novel semantic-enhanced text graph representation learning approach through transformer paradigm. *Cybernetics and Systems, 54*(4), 499–525. https://doi.org/10.1080/01969722.2022.2032795

[20] Zhang, H., & Zhang, J. (2020). Text graph transformer for document classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 8322–8327). https://doi.org/10.18653/v1/2020.emnlp-main.670

Website:

[21]https://www.researchgate.net/publication/354994031_Graph_Convolutional_Network_for_Word_Sense_Disambiguation