Text Simplification Using Graph Attention Networks on BERT Embeddings

April 6, 2025

Abstract

This paper introduces a method for simplifying text using Graph Attention Networks (GATs) applied to BERT embeddings. The goal is to simplify complex semantic representations while preserving their original meaning. I apply this method across two levels of simplification (advanced to intermediate and advanced to elementary) and compare it to a baseline.

To convert the model's output embeddings back to text, I use a simple and efficient nearest neighbor retrieval approach. This helps us avoid more resource-heavy generative models while still producing fluent and understandable output.

I tested the proposed method using the OneStopEnglish dataset, which includes parallel texts at different reading levels. Both numerical results and real examples suggest that my approach is effective at capturing semantic meaning during simplification, making it useful for educational and accessibility-focused applications.

1 Introduction

Text simplification aims to transform complex text into more accessible versions while preserving the original meaning. This task is particularly important for education, accessibility for readers with cognitive disabilities, and language learning. Traditional approaches to text simplification have focused on rule-based systems or sequence-to-sequence models that operate directly on text.

In this paper, I explore a different approach: simplifying text representations in the embedding space using Graph Attention Networks (GATs). By working with BERT embeddings, I take advantage of the rich semantic information captured by these embeddings while leveraging GATs' ability to learn complex transformations between embeddings.

To regenerate readable text from the simplified embeddings produced by the GAT, I introduce a lightweight decoding mechanism based on nearest neighbor retrieval. Instead of using a generative model, I match transformed embeddings to the closest human-written simplified texts in the OneStopEnglish dataset. This enables the pipeline to produce fluent and meaningful output while remaining computationally efficient.

2 Related Work

Text simplification has evolved from rule-based methods to deep neural models. Early approaches used syntactic and lexical rules to restructure complex sentences, as demonstrated by Siddharthan [1]. However, these systems were limited in scalability and generalization.

Statistical machine translation (SMT) reframed simplification as monolingual translation. Wubben et al. [2] proposed a phrase-based SMT model with re-ranking, which improved fluency but required large parallel corpora and suffered from data sparsity.

Sequence-to-sequence models using LSTM architectures [3] became dominant with the rise of deep learning. These systems learned simplification patterns directly from data but sometimes produced semantically inaccurate outputs. To improve controllability, Martin et al. [4] introduced models allowing fine-grained control over sentence length or vocabulary, though they required additional constraints and supervision.

The introduction of BERT [5] marked a shift towards contextual embeddings that capture both syntax and semantics. These have been used successfully in multiple downstream tasks. For simplification, Sun et al. [6] showed that BERT embeddings enhanced alignment and substitution steps in existing pipelines. However, most BERT-based simplification still occurs at the token level, without leveraging structural context.

Graph-based representations offer an alternative to linear sequences by modeling documents as graphs of tokens or sentences connected by syntactic or semantic edges. Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) have been applied to NLP tasks such as semantic role labeling [8] and dependency parsing [10]. GATs, introduced by Veličković et al. [7], assign varying attention weights to neighbors and have shown effectiveness in document classification [9]. Despite this, GNNs remain underutilized in text simplification, especially in embedding-based approaches. This work addresses this gap by using GATs to directly model semantic transformations between advanced and simplified BERT embeddings.

3 Dataset

3.1 One Stop English Corpus

The experiments use the One Stop English corpus [?], which contains texts at three difficulty levels: advanced, intermediate, and elementary. This parallel corpus is particularly valuable for text simplification research as it provides aligned content at different reading levels created by professional editors.

3.2 BERT Embeddings Preparation

I applied pre-processing techniques before generating BERT embeddings for each text at all three reading levels. Specifically, I used the BERT-base model to generate 384-dimensional embeddings that capture the semantic content of each text. These embeddings serve as the input and target for the GAT model.

4 Methodology

4.1 Problem Formulation

Traditional text simplification is framed as a sequence-to-sequence task, mapping complex sentences x_c to simpler versions x_s in the textual domain. In contrast, I reformulate simplification as a regression problem in a high-dimensional semantic space, using contextualized BERT embeddings.

Let $\mathbf{x}_a \in R^d$ be the embedding of an advanced-level text and $\mathbf{x}_s \in R^d$ the embedding of its simplified counterpart (intermediate or elementary). These embeddings are derived from the BERT [CLS] token, with $d = 384$ in the setup.

I aim to learn a transformation function $f_\theta : R^d \to R^d$ such that:

$$f_\theta(\mathbf{x}_a) \approx \mathbf{x}_s \tag{1}$$

This is achieved by minimizing the Mean Squared Error (MSE) loss:

$$\mathcal{L} = \|f_\theta(\mathbf{x}_a) - \mathbf{x}_s\|_2^2 \tag{2}$$

Modeling f_θ using a Graph Attention Network (GAT), allows the transformation to leverage relationships between embedding instances. Documents form nodes in a graph, with edges based on semantic similarity or shared metadata (e.g., topic, length, vocabulary).

This setup enables the model to learn simplification patterns both locally (per instance) and contextually (via neighbors), supporting future integration with decoders for full text generation in a modular simplification pipeline.

4.2 Graph Attention Network Architecture

The model employs a Graph Attention Network (GAT) to transform contextualized BERT embeddings from advanced-level texts into their simplified counterparts. GATs differ from standard Graph Neural Networks by introducing attention mechanisms that assign varying importance to neighboring nodes during message passing. This is especially important when some neighbors, such as semantically related documents, are more informative than others.

Each document in the dataset is represented as a node in a graph, with its BERT embedding serving as the node feature. Edges between nodes are defined using cosine similarity or shared metadata such as topic or genre, enabling the propagation of contextual signals across semantically related examples.

The input is a feature matrix $X \in R^{n \times d}$, where n is the number of nodes and $d = 384$ is the dimensionality of the BERT embeddings. Each row x_i corresponds to an embedding of an advanced-level text.

The first GAT layer applies multi-head attention. For each attention head k, the model computes attention coefficients $\alpha_{ij}^{(k)}$ for node i and its neighbors j, applies a linear transformation $W^{(k)}$, and aggregates the transformed features:

$$h_i^{(1)} = \|_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} W^{(k)} x_j \right)$$

where $\|$ denotes concatenation and σ is a non-linear activation function such as ELU.

The second GAT layer uses a single attention head and averages the outputs from the previous layer, refining the node representation:

$$h_i^{(2)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W x_j^{(1)} \right)$$

The final node embeddings are passed through a feed-forward network with non-linear activation and dropout, projecting them into the simplified embedding space:

$$\hat{\mathbf{x}}_s = \mathrm{FFN}(h_i^{(2)})$$

This architecture allows the model to learn context-aware transformations between embedding spaces. The use of attention enhances generalization by focusing on the most informative semantic neighbors during simplification.

4.3 Training Procedure

I trained two separate Graph Attention Network (GAT) models for text simplification: one mapping advanced-level embeddings to intermediate-level embeddings, and another to elementary-level embeddings. Each model was trained independently on aligned pairs from the One Stop English corpus.

For each pair, the input is a BERT embedding of the advanced text, and the target is the corresponding simplified embedding. These pairs form the node features in a graph, with edges based on cosine similarity or shared metadata (e.g., topic). This structure allows the GAT to propagate information across related examples and learn context-aware transformation patterns.

The model optimizes a Mean Squared Error (MSE) loss between predicted and target embeddings:

$$\mathcal{L}_{\text{MSE}} = \|f_\theta(\mathbf{x}_a) - \mathbf{x}_s\|_2^2 \tag{3}$$

where f_θ is the transformation function learned by the GAT.

Training used the Adam optimizer (learning rate: 0.001), with a batch size of 32 and a maximum of 100 epochs. Early stopping was applied based on validation loss to prevent overfitting.

I monitored cosine similarity between predicted and ground-truth embeddings as a measure of semantic alignment. Dropout was applied after each GAT and feed-forward layer, and weights were initialized using Xavier initialization. All models were implemented using PyTorch Geometric.

This setup enables the model to learn high-quality simplification transformations in the embedding space, suitable for integration into downstream generation or evaluation pipelines.

5 Evaluation and Results

To assess the performance of the Graph Attention Network (GAT)-based text simplification model, I employ cosine similarity as the principal metric. This measures the semantic alignment between the predicted simplified embeddings and their corresponding ground-truth embeddings. Additionally, I compute a baseline similarity between the original advanced-level embeddings and their simplified targets to establish whether the model offers any improvement in semantic proximity. The improvement metric is defined as the difference between the predicted-target similarity and the baseline similarity, indicating the net semantic gain achieved by the transformation.

Table 1 presents the mean results obtained across all test samples for both simplification tasks: advanced to elementary and advanced to intermediate. For elementary simplification, the average cosine similarity between the predicted and true simplified embeddings was 0.858, while the baseline similarity stood at 0.881. This resulted in a slight negative improvement of -0.023. Similarly, for intermediate simplification, the average similarity was 0.859, with a higher baseline similarity of 0.911, yielding a more pronounced negative improvement of -0.052. These figures suggest that while the model is capable of producing embeddings in the correct semantic region of the space, it occasionally overshoots or misaligns the transformation, especially when the source and target texts are already semantically similar.

Target Level	Cosine Similarity	Baseline Similarity	Improvement
Elementary	0.858	0.881	-0.023
Intermediate	0.859	0.911	-0.052

Table 1: Mean evaluation metrics for elementary and intermediate GAT models.

The distribution of cosine similarities is visualized in Figure 1 and Figure 2(Appendix), which show a right-skewed pattern for both tasks. Most predicted embeddings cluster around the 0.85–0.95 range in similarity to their ground-truth counterparts, which reinforces the model's tendency to maintain semantic relevance despite occasional minor regressions.

A closer qualitative examination reveals further insights into model behavior. One representative example from the dataset involves the text titled `Anita`. For elementary simplification, the GAT model achieved a cosine similarity of 0.9438 between the predicted and target embeddings, improving substantially over the baseline similarity of 0.7758. This corresponds to an absolute improvement of approximately 0.168, suggesting a successful transformation that brings the output much closer to the intended simplified representation. The same source text was also evaluated under the intermediate simplification setup, where the model achieved a similarity of 0.9422, up from a baseline of 0.8604, indicating a more subtle but still positive improvement of around 0.082. These results highlight the model's capacity to adapt to different levels of simplification granularity.

Despite these encouraging cases, error analysis reveals several recurring patterns where the model underperforms. Texts containing domain-specific or technical vocabulary often pose a challenge. Such content may be underrepresented in the training data or lack sufficient neighborhood support within the graph, making accurate simplification more difficult. Structural complexity also contributes to failure cases. Sentences with multiple clauses, embedded structures, or idiomatic expressions may be poorly captured by the BERT [CLS] embedding, which tends to compress rich syntactic information into a single vector. The GAT model, lacking hierarchical sensitivity, may not resolve such patterns effectively.

Another issue stems from what I interpret as "overcorrection". In cases where the advanced and simplified versions are already semantically close—particularly in intermediate-level pairs—the model sometimes over-adjusts the embedding, thereby reducing its proximity to the actual simplified target. This trend is reflected in the more negative average improvement observed for intermediate simplification. These observations suggest a potential benefit in incorporating more detailed structural representations within the graph, such as sentence-level or syntactic edge features, to provide richer guidance during transformation.

Overall, the evaluation demonstrates that the GAT model is capable of learning meaningful simplification trajectories in the embedding space. While average improvements are modest, the model consistently produces semantically relevant embeddings that are comparable to human-written simplified representations. Future improvements may focus on refining edge definitions, integrating hierarchical features, and incorporating mechanisms to prevent over-correction when little simplification is required.

6 Discussion

The results from both the quantitative and qualitative evaluations highlight the potential of Graph Attention Networks in learning embedding-level transformations for text simplification. Although the average cosine similarity between predicted and target embeddings remains high for both simplification levels, the mean improvement scores suggest a more nuanced performance. In the case of elementary simplification, the model demonstrates a modest average improvement, and a majority of cases show positive movement toward the target embeddings. By contrast, the intermediate-level model, despite achieving slightly higher average similarity, results in a negative mean improvement and a lower proportion of successful adjustments.

This discrepancy may be due to the reduced semantic gap between advanced and intermediate texts, where the GAT transformation can inadvertently overcorrect an already close embedding. The relatively low standard deviation in cosine similarity for both models suggests consistent predictions, though some outliers do exist—often corresponding to domain-specific vocabulary or texts with abstract phrasing. These instances likely challenge the model due to limited neighborhood information in the graph or limitations in the [CLS] token's ability to summarize complex structures.

Further, the embedding space does not explicitly capture readability features, sentence structure, or lexical simplicity, which are critical for evaluating text simplification in human terms. This explains why embeddings that are semantically aligned may still yield simplified texts that are not clearly easier to read. Nonetheless, the GAT approach remains valuable for producing smooth transformations across the semantic space and offers an efficient route for simplification that avoids expensive autoregressive decoding.

7 Future Work

The current pipeline decodes simplified embeddings using nearest neighbor retrieval, which limits output diversity and confines the system to previously seen human-written examples. A promising future direction is to replace this retrieval mechanism with a generative sequence-to-sequence (Seq2Seq) model that can produce natural language directly from simplified embeddings.

Such a model would be conditioned on the output of the GAT and trained to reconstruct the associated simplified text. This setup, already supported by the existing aligned dataset, could employ a transformer or recurrent decoder, using the embedding as either the initial hidden state or an external conditioning vector.

A generative decoder would allow for evaluation with text-level metrics such as SARI, BLEU, and Flesch-Kincaid, while enabling greater control over output structure and style. Although generation introduces challenges in fluency and semantic alignment, it would significantly enhance the expressiveness and flexibility of the system.

Overall, transitioning to a generative decoder could evolve the current retrieval-based framework into a fully generative and adaptive text simplification model.

8 Conclusion

A different approach to text simplification that operates entirely in the embedding space was introduced. By training Graph Attention Networks on aligned BERT embeddings from advanced and simplified texts, the model learns semantic transformation as a vector mapping task. This enables simplification without relying on explicit text generation.

The nearest neighbor retrieval decoder efficiently translates simplified embeddings back into human-written sentences, offering a lightweight alternative to generative models. This strategy preserves fluency and coherence while significantly reducing computational cost.

Experimental results on the OneStopEnglish dataset show that the model achieves strong semantic alignment with ground-truth simplifications, especially at the elementary level. While intermediate-level gains were more modest, the overall consistency of semantic preservation demonstrates the effectiveness of embedding-level simplification.

These findings validate the potential of GAT-based models in embedding transformation tasks and pave the way for future integration with generative decoders. With further enhancements, this approach could serve as a scalable, accurate, and interpretable component in simplification pipelines for educational and accessibility-focused applications.

9 References

References

[1] Siddharthan, A. (2006). Syntactic simplification and text cohesion. Research on Language and Computation, 4(1), 77-109.

[2] Wubben, S., Van Den Bosch, A., & Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.

[3] Nisioi, S., Štajner, S., Ponzetto, S. P., & Dinu, L. P. (2017). Exploring neural text simplification models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.

[4] Martin, L., de la Clergerie, É., Sagot, B., & Bordes, A. (2020). Controllable sentence simplification. In Proceedings of the 12th Language Resources and Evaluation Conference.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT.

[6] Sun, T., Shao, Y., Qiu, X., Guo, H., Hu, J., & Huang, X. (2021). BERT-based text simplification. In Findings of the Association for Computational Linguistics.

[7] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In International Conference on Learning Representations.

[8] Marcheggiani, D., & Titov, I. (2018). Encoding sentences with graph convolutional networks for semantic role labeling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.

[9] Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence.

[10] Ji, T., Wu, Y., & Lan, M. (2019). Graph-based dependency parsing with graph neural networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

[11] Vajjala, S., & Lučić, I. (2018). OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 297–304).

Appendix

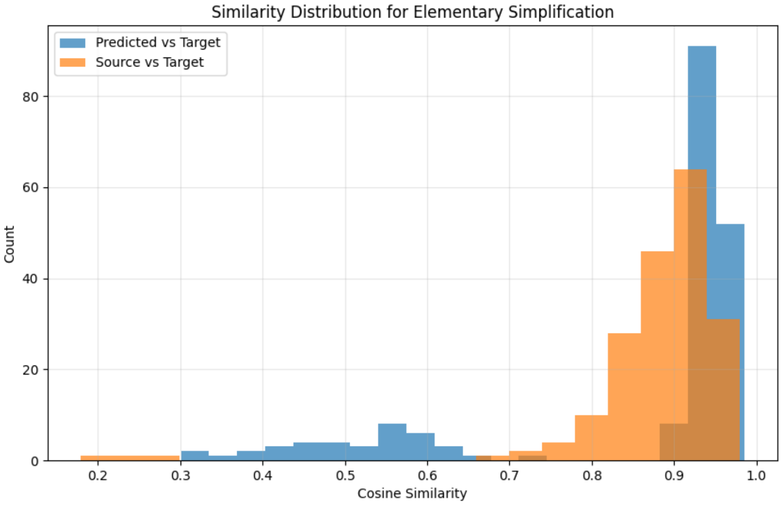

Figure 1: Cosine similarity distribution between predicted and target embeddings for the elementary-level GAT model.

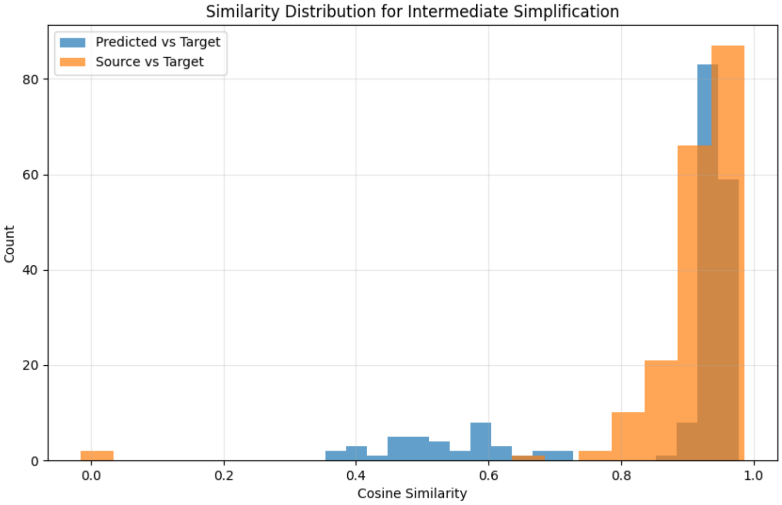

Figure 2: Cosine similarity distribution between predicted and target embeddings for the intermediate-level GAT model.

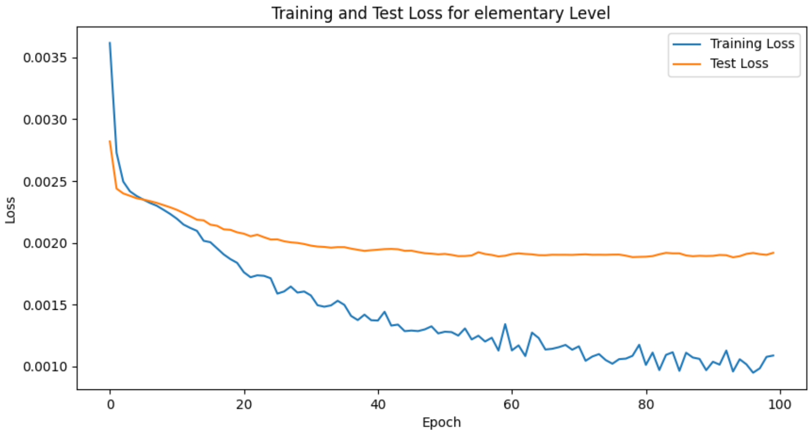

Figure 3: Training loss over epochs for the GAT model mapping advanced to elementary embeddings.

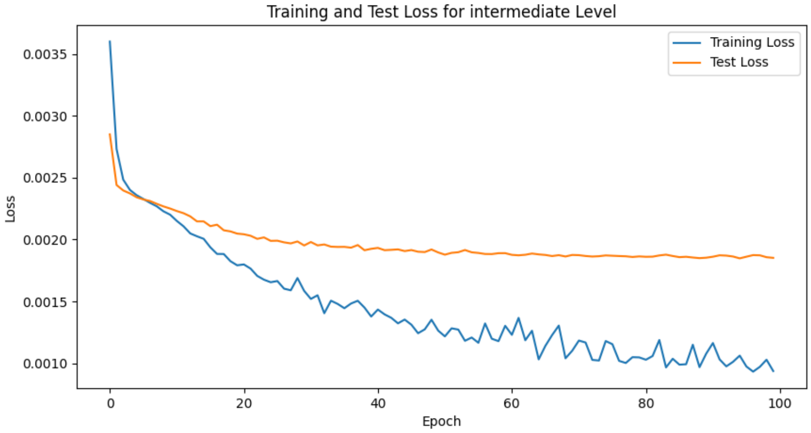

Figure 4: Training loss over epochs for the GAT model mapping advanced to intermediate embeddings.