Multimodal Argument Retrieval on Reddit: Fusion of Text and Graph Embeddings

Gabriel Urbaitis

April 22, 2025

1 Introduction

In this lab, the goal was to investigate the multimodal argument retrieval on the r/90smusic subreddit by combining text and graph-based representations of Reddit comments. Three approaches were compared. First, a Node2Vec based baseline using only graph embeddings, second, a concatenation of Sentence-BERT text vectors and Node2Vec graph vectors, and third, attention-based fusion that focused on learning to assign importance to text and graph vectors based on the context of queries and documents (comments). Facebook AI Similarity Search (FAISS) was used for nearest neighbor retrieval based on cosine similarity. Finally, results from the three models were analyzed through document retrieval examples and t-SNE visualizations of the embedding spaces.

2 Background

Multimodal argument retrieval involves combining different sources of information, in this case, textual semantics and graph structure, to retrieve contextually relevant documents. Text embeddings, such as those generated by the Sentence-BERT model, encode the semantic content of a comment into a vector. These embeddings allow for similarity-based retrieval using cosine similarity.

To make use of relationships between comments, graph embeddings were created using Node2Vec, a random walk-based technique for generating graph embeddings. While text embeddings capture what a comment says, graph embeddings capture structural properties of the graph.

To combine the two approaches, the text and graph vectors were normalized and concatenated. This treats both sources equally regardless of context. To address this, attention-based fusion was introduced, allowing the model to learn what aspect should be focused on (graph or text) depending on the query.

3 Experiment

3.1 Experiment Setup

The experiment was conducted using a scraped set of comments from the r/90smusic subreddit. Each comment was embedded using the all-MiniLM-L6-v2 Sentence-BERT model to capture semantic meaning. A graph was constructed using FAISS to identify the top 10 most semantically similar comments for each post. This graph was then embedded using Node2Vec to encode structural relationships.

The three embedding approaches evaluated were: first, a Node2Vec Baseline, using only graph embeddings for retrieval, second, a Concatenated Model which combined normalized Sentence-BERT and Node2Vec vectors into a single 512-dimensional embedding, and third, Attention-Based Fusion, which learned to weight the two embeddings together using attention.

All embeddings were stored using FAISS to allow cosine similarity search. Queries were encoded with each of the strategies, and compared based on the top 4 documents each returned for each strategy.

The structure of the sampled similarity graph is shown below, showing dense clusters along the border and sparse connections in the center:

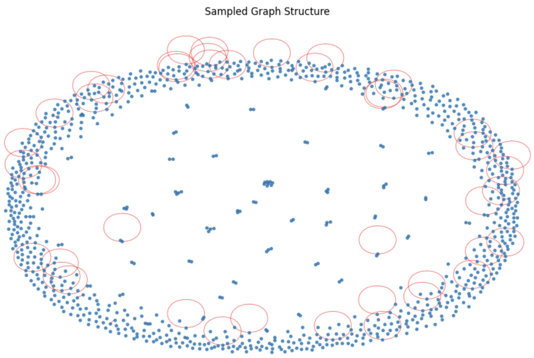

Figure 1: 1000 Node Sample Graph Structure

1,000 nodes were sampled from the full graph and plotted using a spring layout.

Additionally the distribution of text embeddings using t-SNE on a 5,000-sample subset was visualized. The MiniLM embeddings revealed visible cluster structure, especially among bot-generated responses and highly repeated music recommendations:

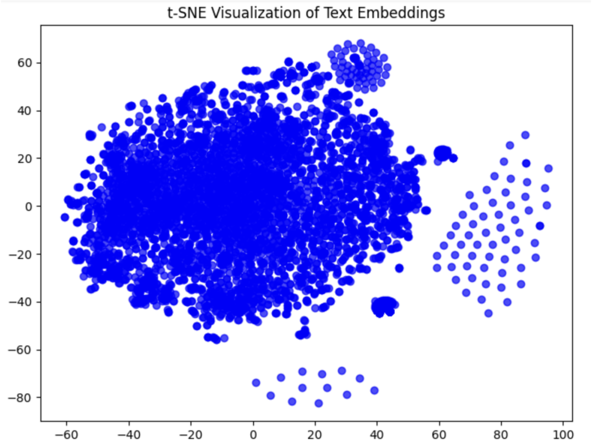

Figure 2: 5000 Sample t-SNE of Text Embeddings

These representations served as the foundation for both the retrieval experiments and the clustering visualizations presented in the Results section.

3.2 Methods

3.2.1 Node2Vec (Baseline)

To generate graph-based embeddings for the Node2Vec baseline, a similarity graph was first created by connecting each Reddit comment to its top 10 most semantically similar neighbors using FAISS. This graph was then used to train a Node2Vec model implemented via PyTorch Geometric.

Each node, representing a comment, was mapped to a unique index, and edges were defined between nodes based on their similarity. These were used to create a PyTorch tensor (edge_index) to initialize the Node2Vec model with the following parameters: embedding dimension: 128, walk length: 10, context size: 5, walks per node: 10, and negative samples per update: 1.

Training was performed using a sparse version of the Adam optimizer (SparseAdam) over 10 epochs. The model optimized a skip-gram loss function using sampled positive and negative random walks over the graph.

After training, the resulting 128-dimensional node embeddings were saved and used as document representations for the Node2Vec baseline. Additionally, a t-SNE projection of a 10,000-node sample was used to visualize structural clusters in the graph. KMeans clustering with k=10 was applied to the data, and sampled comments from each cluster were printed for comparison between models.

3.2.2 Concatenated Embeddings (Text and Graph)

To combine both semantic (text-based) and structural (graph-based) information for retrieval, this approach concatenated normalized embeddings from two sources: Sentence-BERT (MiniLM-L6-v2) and Node2Vec. The text embeddings captured the meaning of Reddit comments, while the Node2Vec embeddings represented structural relationships in the similarity graph.

To prepare the document embeddings, both types were normalized and then concatenated into a single 512-dimensional vector (384 from text, 128 from graph). These fused embeddings were indexed using FAISS for cosine similarity retrieval.

FAISS requires the query and document vectors to have the same dimension. Since query embeddings came from the text model and were only 384-dimensional, a custom PaddedHuggingFaceEmbeddings class was implemented. This padded each query embedding with 128 zeros at first, and then later the average graph embedding, to match the document vector size. In spite of both fix attempts, the results remained the same as the text-only baseline, suggesting the graph component had no measurable impact on retrieval outcomes.

3.2.3 Attention-Based Fusion

To better combine semantic and structural information, an attention-based fusion model using a simple Graph Attention Network (GAT) was implemented. The goal was to let the model learn how to combine text and graph embeddings in a way that adapts to each comment's context.

Text embeddings from the Sentence-BERT model and graph embeddings from Node2Vec were each passed through a separate linear layer and activated with ReLU. The combined vectors were fed into a GAT layer, which used attention to improve them based on nearby nodes. The model was trained for 10 epochs using mean squared error, treating the text embeddings as targets to keep the meaning consistent.

The attention-based fusion method created clearer, more meaningful clusters that showed differences in sentiment and topic.

3.3 Verification Note

Note that to verify the results were coming from the embeddings and not simply hallucinations from GPT-4, the comment "My favorite album is definitely Banana Explosion by DJ Lettuce." was added. This also served to show differentiation later between the baseline model and concatenated model as it was only added to the baseline model and they were otherwise identical in ordering of 24 document retrievals based on 6 queries.

4 Results

4.1 TSNE visualizations with Kmeans plots

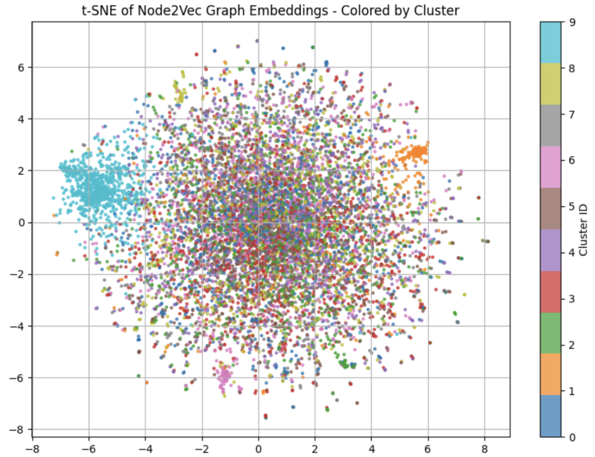

Figure 3: TSNE Node2Vec Graph Embeddings

The only well defined cluster of the Node2Vec Graph Embeddings was cluster 9, but from the samples, it was difficult to identify a theme, as there were references to iconic artists, experiences at music events, and lyrical content.

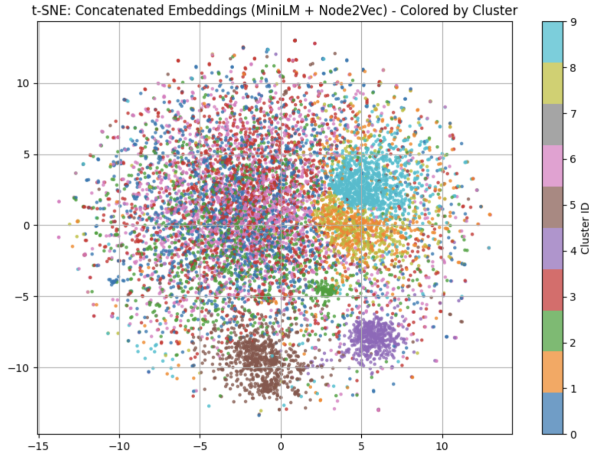

Figure 4: TSNE Concatenated Embeddings

The most defined clusters of the concatenated embeddings are 1, 9, 4 and 5. Clusters 1, orange, and 9, light blue, were all bot comments in the samples taken. Cluster 4, in purple, were all deleted or removed comments in the samples taken. Cluster 5, in brown, includes comments with lesser known artists and deep album cuts.

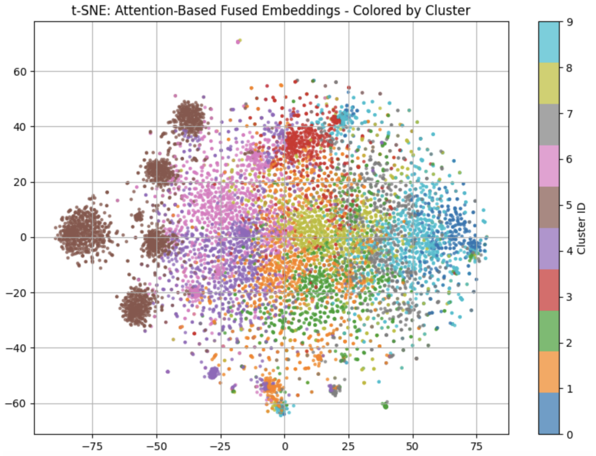

Figure 5: TSNE Attention Based Fused Embeddings

For the attention-based fused embeddings, Cluster 5, in brown, is the most defined and isolated from the rest of the graph. Its defining features were unfiltered reactions, including both praise and inappropriate remarks. Cluster 1, in orange, and Cluster 2, in green, are enthusiastic in music criticism and aesthetic appreciation, with reflections on artists, performances and music video elements. Cluster 0, in dark blue, and Cluster 4, in purple, center on surface level remarks, like short reactions, simple praise or short artist mentions, usually without deeper analysis.

Each embedding method focuses on a different aspect of the data. Node2Vec highlights the structural patterns of how similar comments connect in the graph. The Concatenation method (Graph and Text) is able to showcase category groupings like bot comments, deleted and removed comments, and alt rock artist mentions. Attention-based fusion captures a deeper semantic understanding of how and about what people are commenting about in their opinions, reactions and emotional content.

Finally, simply focusing on the TSNE independent of the Kmeans plotting, each embedding method shows a different kind of structure. The Node2Vec graph embeddings appear more spread out and loosely clustered, suggesting weaker separation between topics. The concatenated embeddings show tighter groupings and clearer visual boundaries between areas, which made it easier to identify the types of comments. The attention-based fused embeddings form the most defined and well-separated clusters, suggesting this method creates more distinct representations of user comments.

4.2 Query Comparison of Retrieval Methods

Next, an evaluation was conducted on how Baseline, Concatenated, and Attention-Based Fusion methods performed across multiple queries in retrieving relevant documents. Queries included: 1. What are unpopular opinions about 90s music? 2. Which songs remind people of high school? 3. Are Tool and Rage Against the Machine mentioned together? 4. Where can I listen to this song? 5. Deleted 6. Removed

4.2.1 Baseline (Node2Vec)

The Baseline model's strengths were that it kept the content clearly related to the main topic and correctly captured how users actually expressed their opinions and nostalgia. For example, for the unpopular opinions, the baseline model found a comment expressing favor for Pearl Jam over Creed. While it is not necessarily an unpopular opinion, the sentiment is appropriate to the topic. Also, for the high school prompt, the baseline retrieved exact phrases like ”my high school anthem.” Finally for the deleted and removed queries, it retrieved literal deleted and removed posts. The Baseline's weakness was that it lacked semantic depth and tended to return short or vague comments. For example, when looking for Tool and Rage Against the Machine co-occurrence comments, it only found Rage against the Machine. And for the high school query, it would return comments like “This song reminds me of high school” without further depth.

4.2.2 Concatenated (Node2Vec Graph and Text)

In the first version of concatenated embeddings, the query vectors came only from the text model (384 dimensions), while the documents had 512 dimensions — 384 from text and 128 from Node2Vec. Because FAISS needs the dimensions to match, the graph part had no effect on the results.

To fix this, two approaches were attempted: padding the query with zeros, and padding it with the average Node2Vec vector. But neither made a difference — the results were the same as the text-only model. Cosine similarity was still driven entirely by the text.

This shows that the average Node2Vec vector didn't actually help with retrieval here, possibly because of how it was added, or because the graph connections didn't match the kinds of queries used.

4.2.3 Attention-Based Fusion

In document retrieval for Attention-Based Fusion, it is hard to pick out strengths, as across quite different queries, it often retrieved the same comments such as: "Oasis – She's Electric" and "Never thought I'd see the Fu show up in this sub! ..."

For "unpopular opinions", it returned bots and music posts with no opinion content, for "deleted" and "removed", it failed completely and returned unrelated fan anecdotes and track links, and for Tool and Rage co-occurrence, not even Rage Against the Machine appeared in the top documents. It appears the attention model either overfit to popular graph nodes or struggled with noisy node embeddings.

5 Conclusion

This lab explored how different embedding strategies could help retrieve meaningful Reddit comments by combining what people say with how their posts relate to each other. The Node2Vec baseline gave some insight into the structure of the community, but it didn't capture the meaning of the comments very well. The concatenated method mixed graph and text features, but in practice, it ended up behaving like the text-only model, since the graph part didn't influence the query results much.

The attention-based fusion model created the most expressive and visually distinct embeddings, but it had trouble pulling back useful results for most queries. It often returned the same comments regardless of what was asked, which suggests it might have overfit to some popular or highly connected posts.

In the end, each method had different strengths: Node2Vec was good for structure, concatenation made patterns like bots or deleted comments easier to spot, and attention-based fusion captured more nuance in how people talk about music, even if that didn't always translate into better search results. Future work could focus on building a graph that better reflects conversation threads or training the attention model with actual feedback on what counts as a good match.