Sarcasm Detection and Rewriting

Team 38 - The SUS

Umang Patel (2022101037)

Sparsh Goel (2022101051)

Sahil Patel (2022101046)

Introduction

- **Problem:** Sarcasm is prevalent on social media but misleads sentiment analysis algorithms
- **Challenge:** Sarcasm involves a complex structure of:
  - Contextual semantics
  - Syntactic patterns
  - Affective markers
  - Implicit contradictions
- **Impact:** Inaccurate sentiment analysis leads to:
  - Misleading insights from user feedback
  - Incorrect interpretation of customer reviews
  - Flawed brand sentiment measurement
- **Our solution:** A two-stage system that detects sarcasm and transforms it into literal language

Why Is Sarcasm Detection Difficult?

- Inherent irony and contradiction
- Relies on cultural context and shared knowledge
- Often combines positive words with negative intent
- Requires understanding beyond surface-level semantics
- **Examples:**
  - "Congratulations on stating the obvious." → Appears positive but conveys criticism
  - "Sure, let's ignore all evidence." → Seemingly agreeable but implies disagreement
  - "I'm sure glaciers will start moving any minute now." → Expresses disbelief through feigned certainty

Project Objectives

- **Primary Goal:** Develop an end-to-end system for sarcasm detection and rewriting
- **Specific Objectives:**
  - Design an architecture for sarcasm detection
  - Create a robust feature representation that captures semantic, syntactic, and affective aspects
  - Develop a rewriting model that preserves content while removing sarcastic tone
  - Analyze model behavior through visualizations of internal mechanisms

Literature Review: Sarcasm Detection

- **Mohan et al. (2023) - "Sarcasm Detection Using BERT and GCN"**
  - Combined transformer and GCN approaches
  - Created separate dependency and affective graphs
  - Used alternating graph processing
  - Achieved 90.7% accuracy on the Headlines dataset
- **Yaghoobian et al. (2021) – “Sarcasm Detection: A Comparative Study”**
  - Compared various ML and DL techniques
  - Highlighted the importance of context features
  - Showed limitations of pure lexical approaches
- **Šandor and Bagic Babac (2023) – “Sarcasm Detection in Online Comments Using Machine Learning”**
  - Focused on online user comments
  - Explored feature importance in the social media context

Literature Review: Text Attribute Transfer

- **Li et al. (2018) - "Delete, Retrieve, Generate"**
  - Identified attribute markers through frequency analysis
  - Proposed three-step framework:
    - Delete attribute markers from source text
    - Retrieve similar content with the target attribute
    - Generate new text combining content and target style
  - Demonstrated effectiveness on sentiment transfer tasks
- **Our Approach:** We ultimately opted for a GAN-based architecture for the rewriting component, the delete-retrieve-generate paper influenced our decision to use the GAN-based network as simple delete-retrieve-generate won’t be able to capture the contextual relationships

Datasets

- Combined multiple datasets for comprehensive training:
  - **Sarcasm on Reddit:**
    - 1.3 million comments (self-annotated)
    - Diverse range of topics and contexts
    - Contains natural sarcasm expressions
  - **Mustard Dataset:**
    - Sarcastic comments from popular TV shows (Friends, Golden Girls, Big Bang Theory)
    - Includes multimodal information
    - Professional dialogue with deliberate sarcasm
- **iSarcasm Dataset:**
  - 4,484 tweets with 777 labeled as sarcastic
  - Used specifically for training the rewriting module
  - Contains human-written non-sarcastic counterparts for evaluation

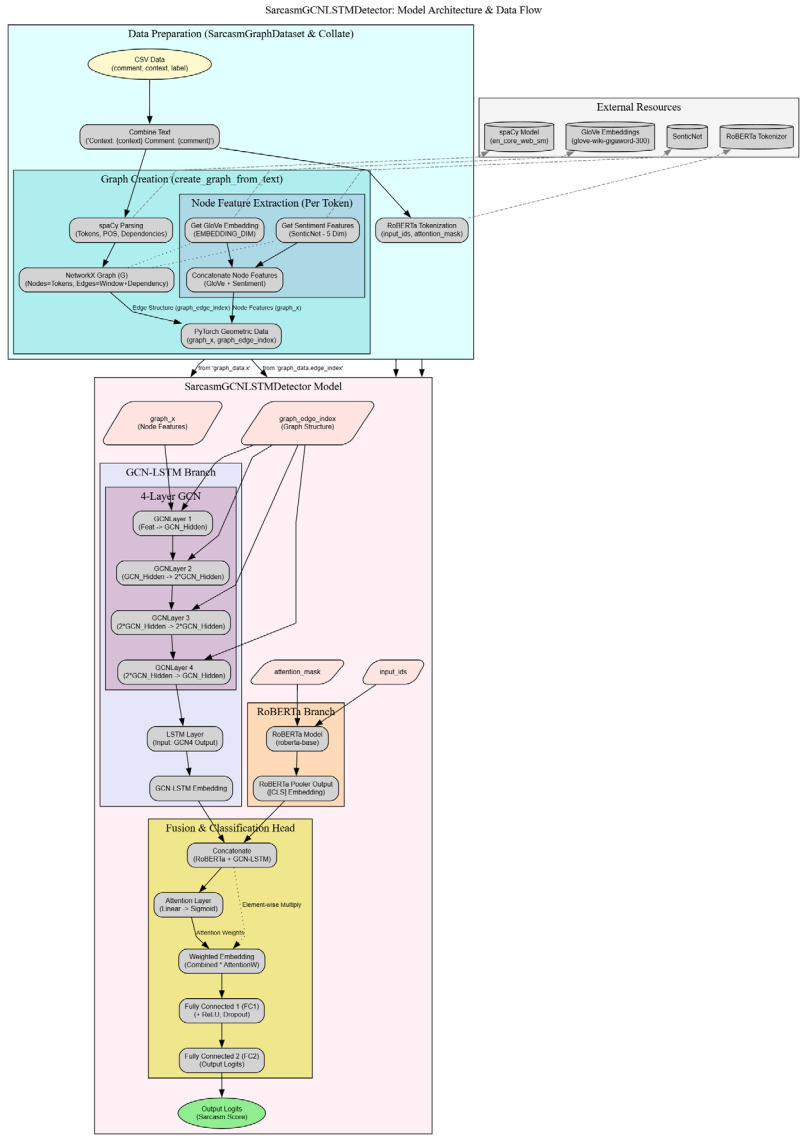
Data Preprocessing & Feature Engineering

- **Preprocessing Pipeline:**
  - Tokenization using NLTK and spaCy
  - Lowercasing and punctuation normalization
  - Lemmatization for word normalization
  - Context extraction for relevant samples
- **Feature Extraction:**
  - **Semantic Features:**
    - 300-dimensional GloVe embeddings (glove-wiki-gigaword-300)
    - Captures distributed word semantics
  - **Affective Features:**
    - 5-dimensional sentiment vector from SenticNet
    - Components:
    - Polarity value (-1 to 1)
    - Positive sentiment flag
    - Negative sentiment flag
    - Neutral sentiment flag
    - Intensity value

Graph Construction: Technical Details

- **Node Creation:**
  - Words extracted using the spaCy dependency parser
  - Each node represents a token in the text
  - Node features: [GloVe embedding (300D) || SenticNet vector (5D)]
- **Edge Formation:**
  - Window-based Edges:
    - Connect adjacent words within a window size of 2
    - Capture local context and n-gram relationships
    - Help the model learn phrase-level patterns
  - Dependency-based Edges:
    - Derived from a syntactic dependency parse
    - Represent grammatical relationships
    - Enable long-distance connections between related words
- **Implementation:** PyTorch Geometric framework with NetworkX for initial graph creation

Detection Model Architecture: Overview



Detection Model Architecture: Overview

- **Hybrid Architecture Components:**
  - **Text Encoder**: RoBERTa extracts contextual embeddings
  - **Graph Processing**: 4-layer GCN with sequential modeling
  - **Feature Fusion**: The Attention mechanism combines representations
  - **Classification**: Fully connected layers for final prediction

- **Innovation**: Integrates transformer-based language understanding with graph-based structural and sentiment analysis

Detection Model: RoBERTa Component

- **Transformer-based Text Encoding:**
  - **Base Model**: RoBERTa-base (125M parameters)
  - **Input Processing:**
    - Special tokenization with BPE
    - Max sequence length: 128 tokens
    - Attention masking for variable-length inputs
  - **Output**: 768-dimensional pooled embedding from [CLS] token
  - **Training**: Fine-tuned with adapter layers to prevent catastrophic forgetting
- **Purpose**: Capture deep contextual relationships and semantic understanding of text, particularly implicit meaning in sarcastic expressions

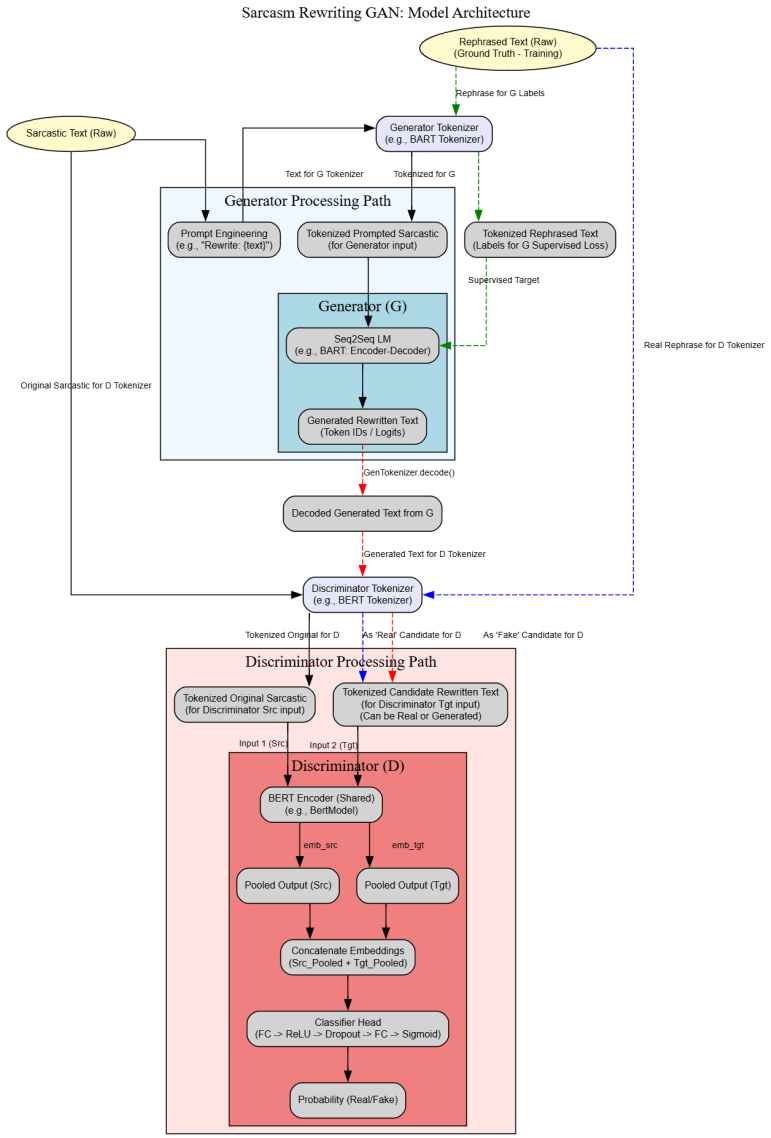
Detection Model: Graph Convolutional Network

- **4-layer GCN Implementation:**
  - Layer 1: Input dim (305) → Hidden dim (64)
  - Layer 2: Hidden dim (64) → Hidden dim (128)
  - Layer 3: Hidden dim (128) → Hidden dim (128)
  - Layer 4: Hidden dim (128) → Hidden dim (64)
- **Technical Details:**
  - GCNConv layers from PyTorch Geometric
  - BatchNorm1d after each convolution
  - ReLU activation and dropout (0.2)
  - Processes the integrated graph with dual edge types
  - Gradually increases and then decreases feature dimensions for the bottleneck effect
- **Purpose**: Learn structural patterns in text that indicate sarcasm, particularly syntactic contradictions and sentiment incongruities

Detection Model: Sequential Processing & Fusion

- **Sequential Processing:**
  - LSTM processes the GCN output sequence
  - Bidirectional with hidden dim 32 (64 total)
  - Captures sequential dependencies in node features
  - For batch processing: mean aggregation strategy
- **Attention-based Fusion:**
  - Combines RoBERTa embedding with GCN-LSTM output
  - Attention mechanism: Linear(hidden_dim + gcn_dim, 1)
  - Dynamic weighting based on feature importance
  - Allows the model to focus on different aspects per sample
- **Final Classification:**
  - FC layer: (hidden_dim + gcn_dim) → 256
  - ReLU activation + dropout
  - Output layer: 256 → 1 (binary classification)
  - Sigmoid activation for probability output

Rewriting Module: GAN Architecture



Rewriting Module: GAN Architecture

- **Generator**: BART encoder-decoder model for text transformation
- **Discriminator**: BERT-based classifier for authenticity evaluation
- **Training Objective**: Combined supervised and adversarial losses
- **Inference Process**: Beam search with width 10 for diverse outputs

Rewriting Module: Technical Details

- **Generator Architecture:**
  - Base Model: BART-base with 140M parameters
  - Encoder: Processes sarcastic input text
  - Decoder: Generates non-sarcastic equivalent with attention on encoder
  - Input Engineering: Special prompt prefixing ("Rewrite without sarcasm:")
- **Discriminator Architecture:**
  - Base Model: BERT-based classifier
  - Input Structure: [CLS] sarcastic text [SEP] rewritten text [SEP]
  - Output: Probability that transformation is authentic (human-like)
  - Architecture: Pooled output → FC → Sigmoid
- **Loss Functions:**
  - **Generator Loss**: $L_G = L_{supervised} + \lambda \cdot L_{adversarial}$
    - $L_{supervised}$: Cross-entropy on token prediction
    - $L_{adversarial}$: Binary cross-entropy from discriminator
  - **Discriminator Loss**: $L_D = 0.5(BCE(D(x,y_{real}), 1) + BCE(D(x,y_{fake}), 0))$

Performance Metrics: Validation Results

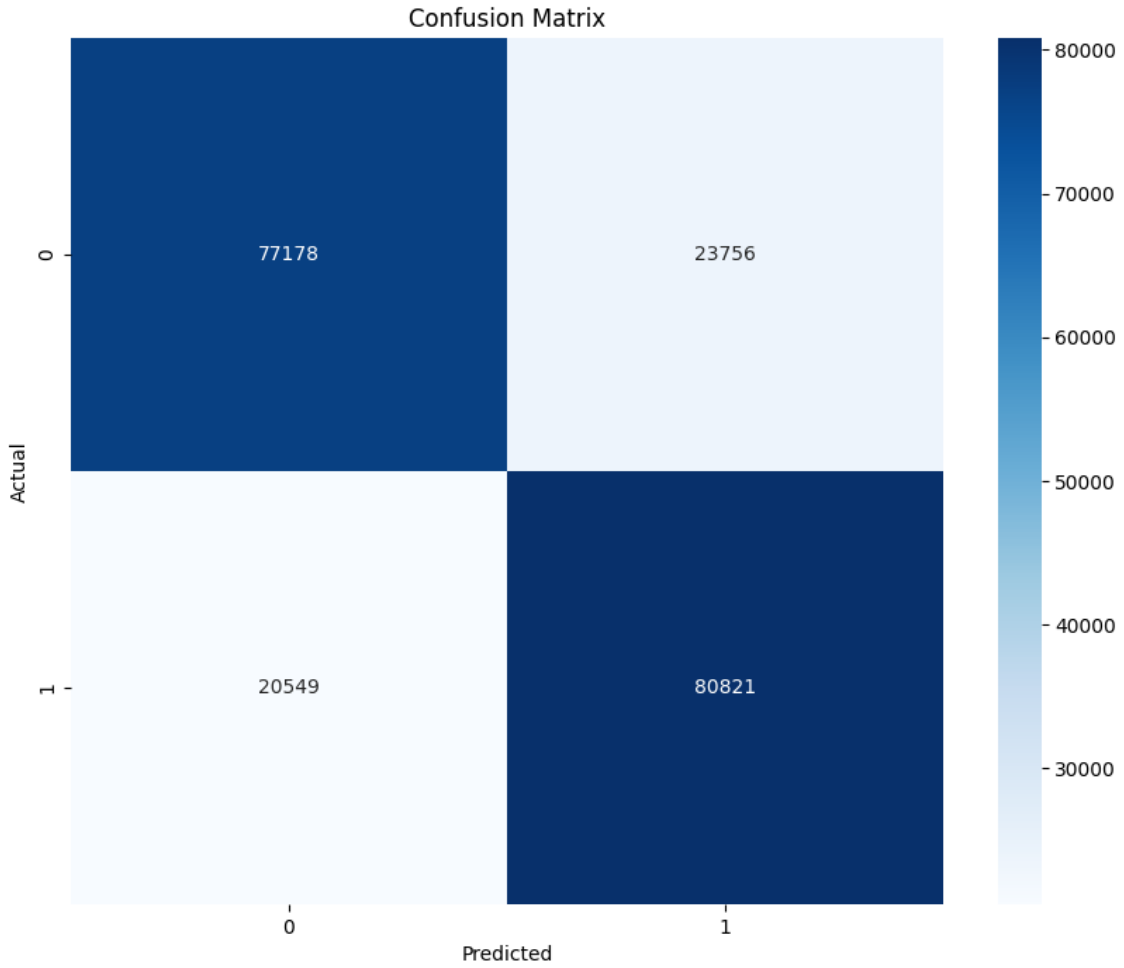
- **Validation Performance Progression:**

Epoch	Accuracy	F1-score
1	79.57%	80.31%
2	83.02%	82.44%
3	85.20%	85.04%
4	85.98%	86.15%

- Consistent improvement across training epochs
- F1-score trajectory aligns with accuracy

Performance Metrics: Test Results

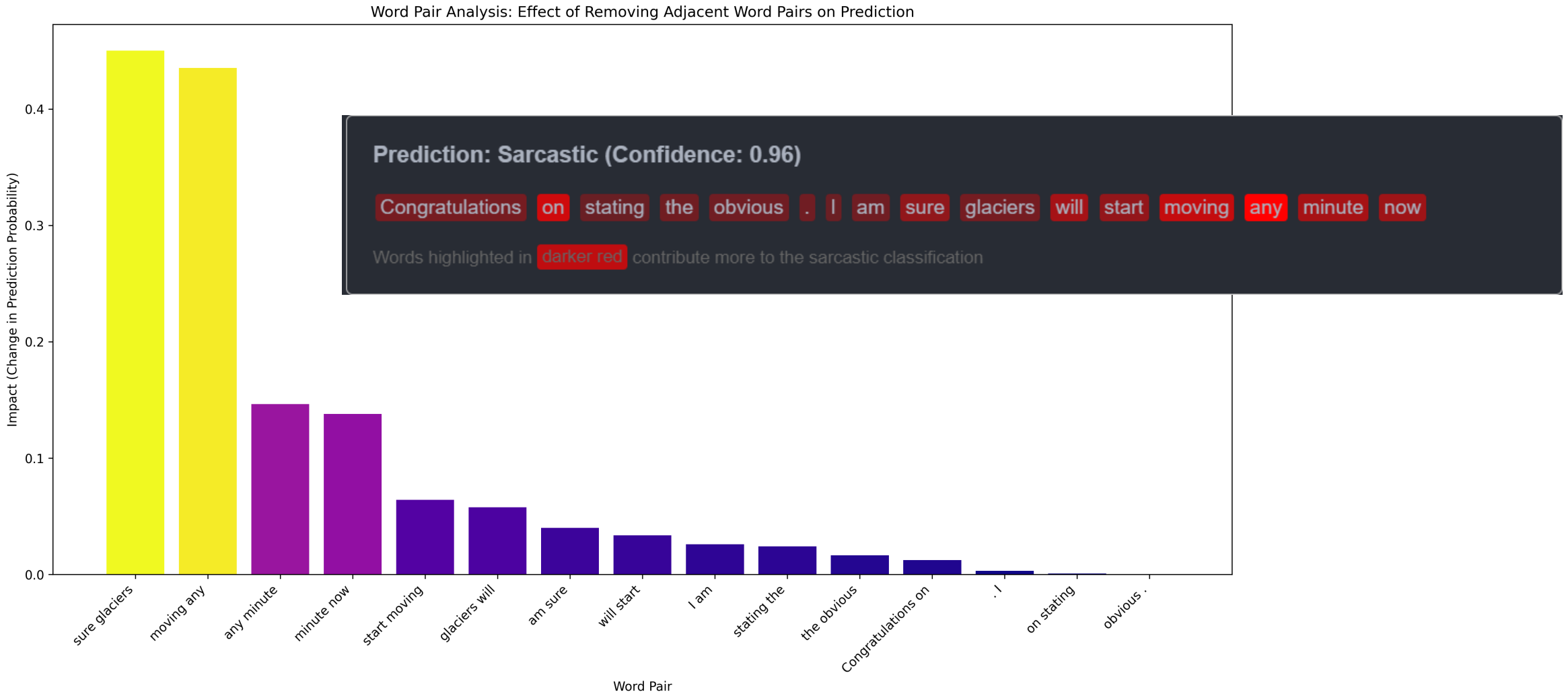
- **Final Test Set Performance:**
  - Accuracy: 78.10%
  - F1-score: 78.49%
  - Precision: 77.43%
  - Recall: 79.58%



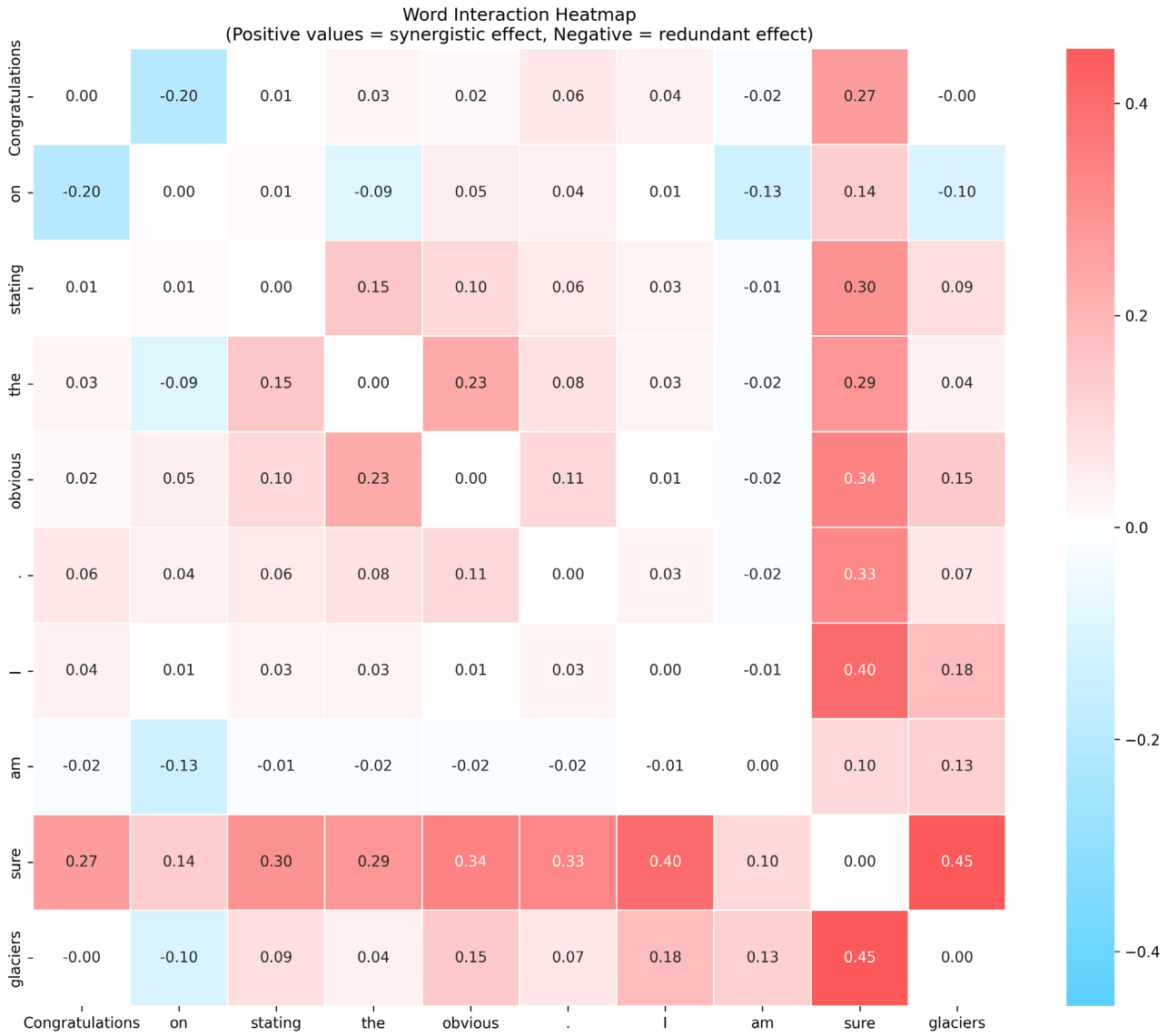
Sarcasm Rewriting Results: Examples

Sarcastic Input	Non-sarcastic Output
Congratulations on stating the obvious. I'm sure glaciers will start moving any minute now	There is no need to stating the obvious.
Nice of you to show up three minutes late your timing really is something else	It's not nice of you to show up three minutes late.
I don't need your approval, darling, I have my own	I don't need your approval, I have my own opinion.
Sure, let's ignore all evidence and cling to your flawless reasoning	There is no need to ignore all evidence and cling to my flawless reasoning.
Absolutely, let's add that to the dozen other things I definitely wasn't planning to do.	There are a lot of things I don't like doing.
Congratulations on arriving exactly three minutes late Your punctuality is truly inspiring	Having to arrive exactly three minutes late is disappointing but not disappointing.

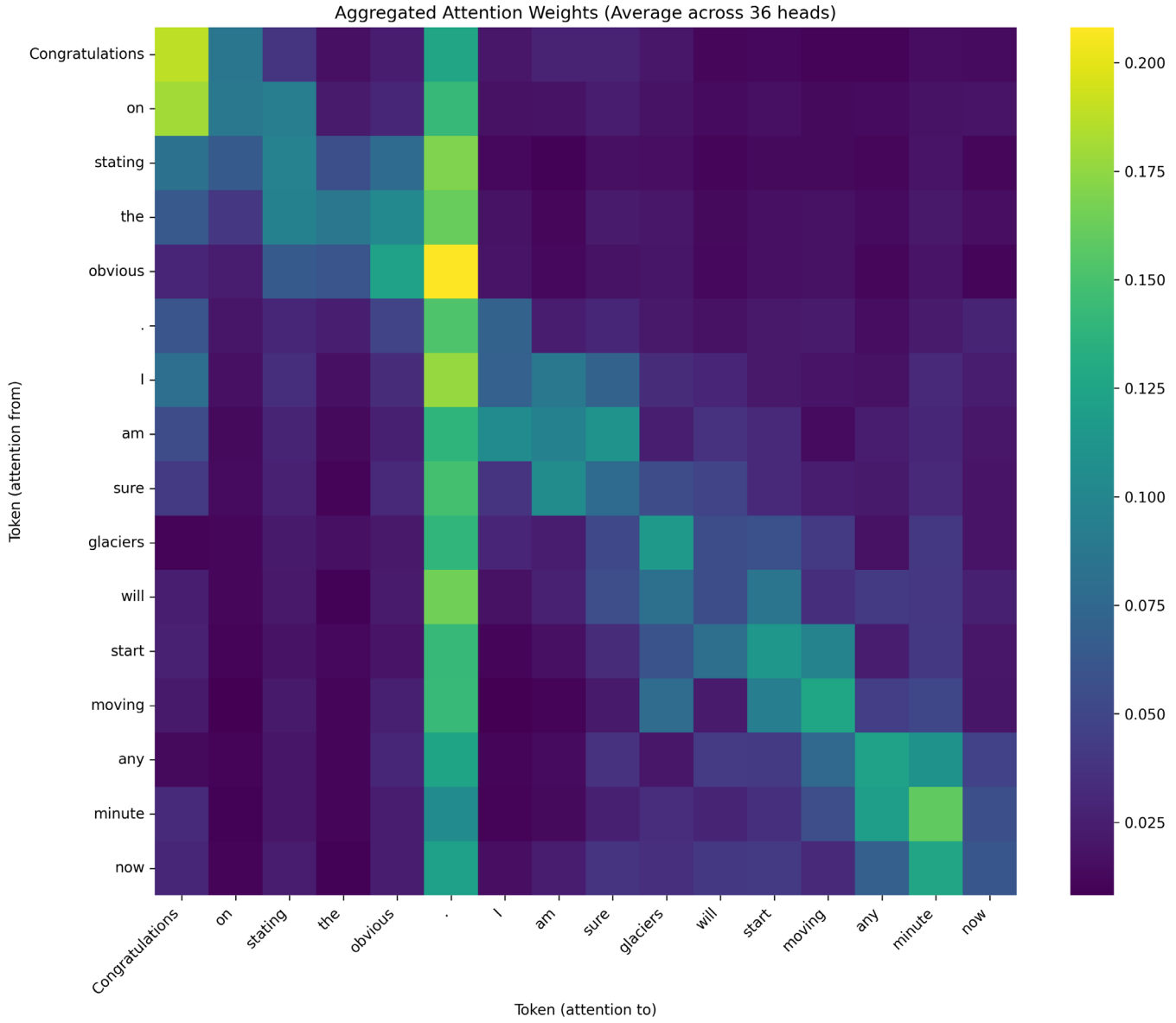
Word-Level Importance Analysis



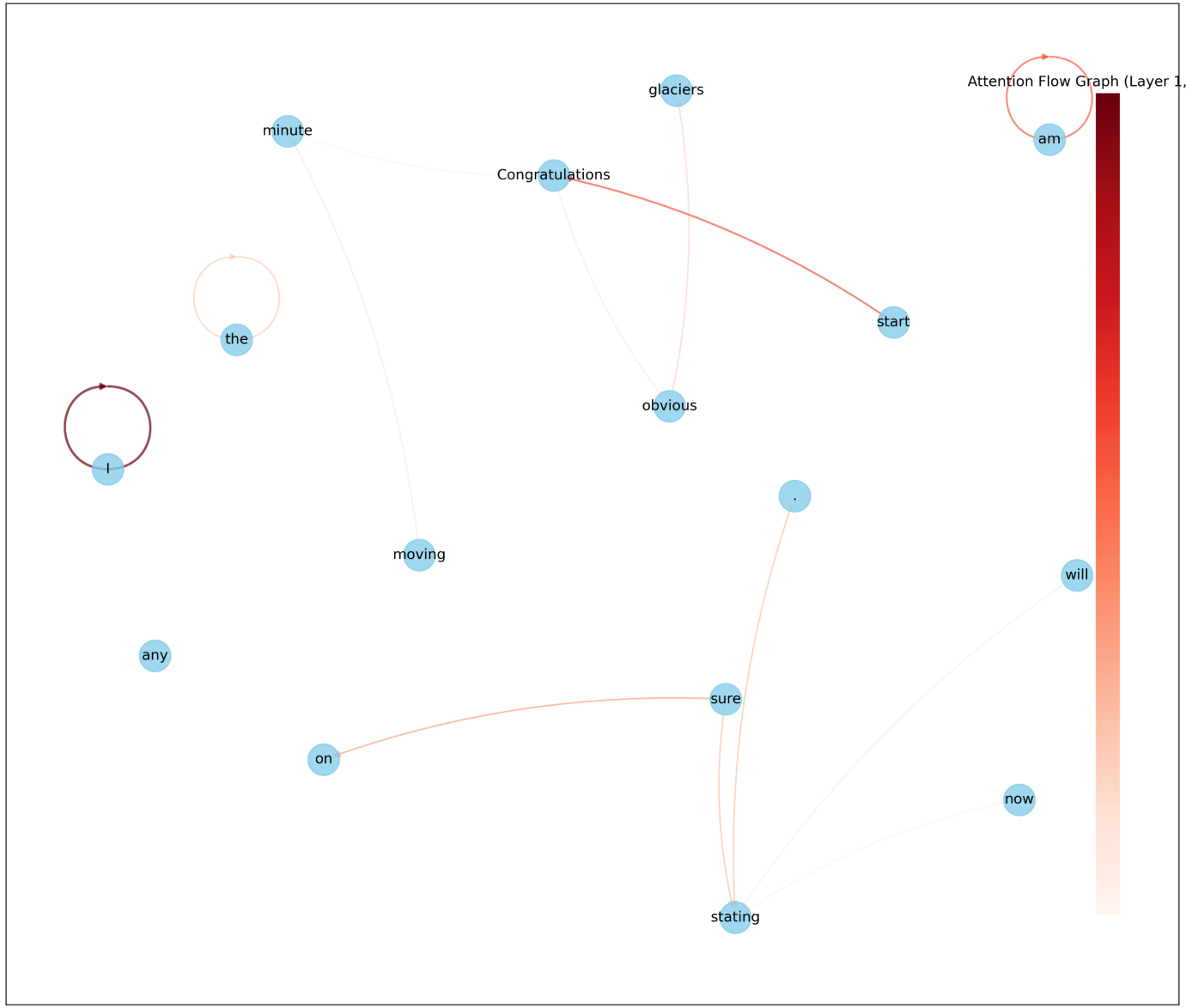
Word-Level Importance Analysis



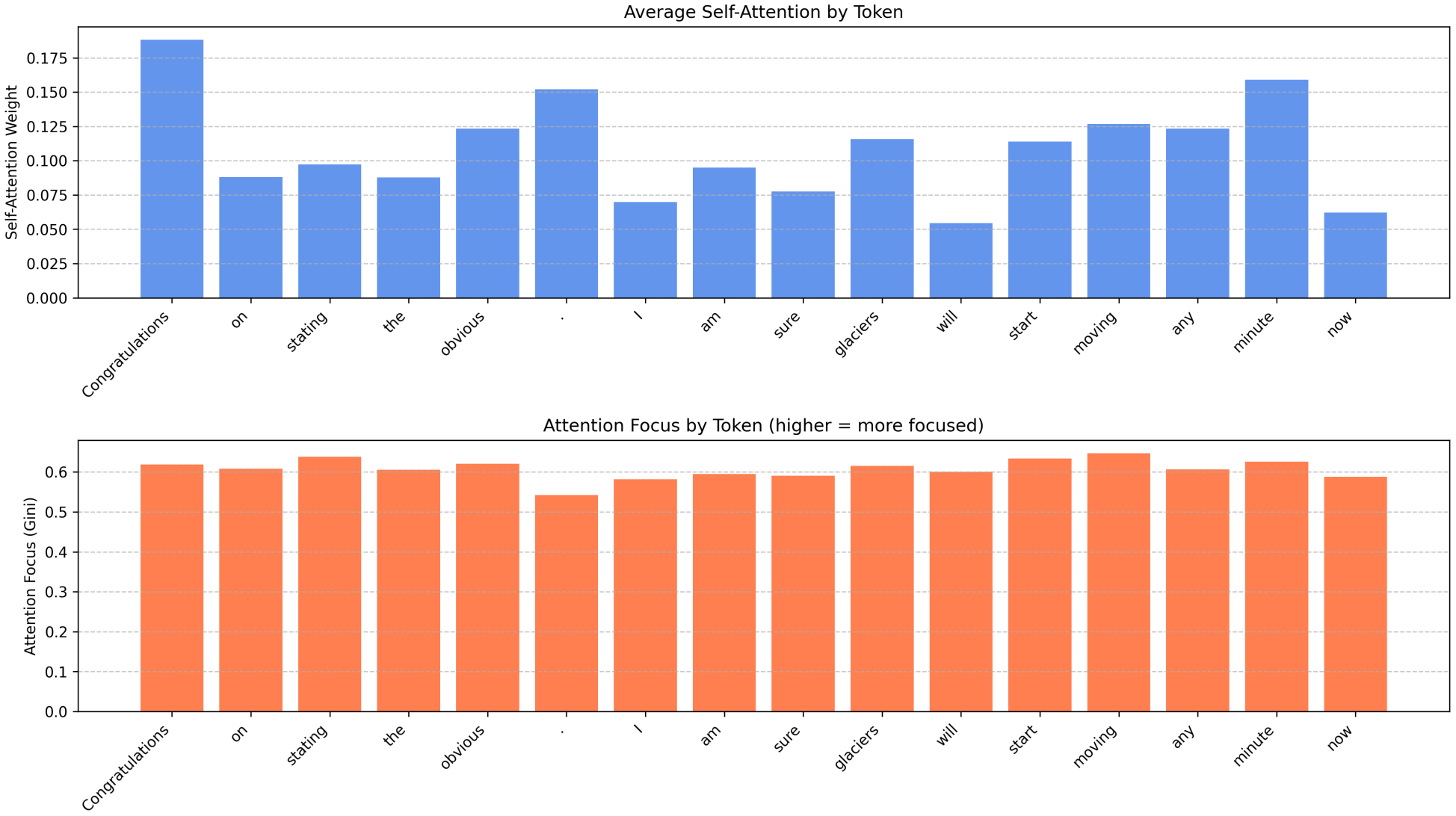
Attention Mechanism Analysis



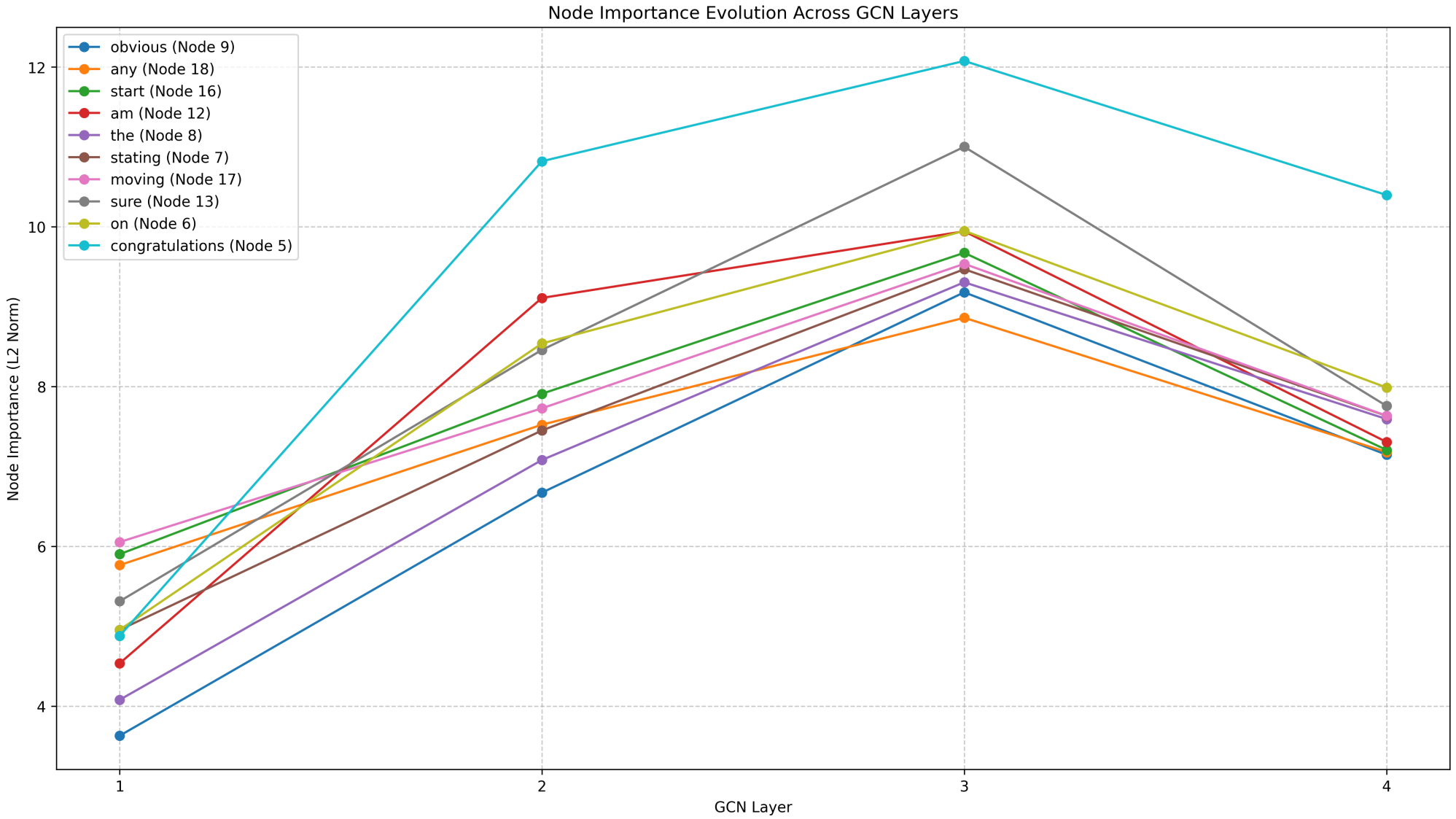
Attention Mechanism Analysis



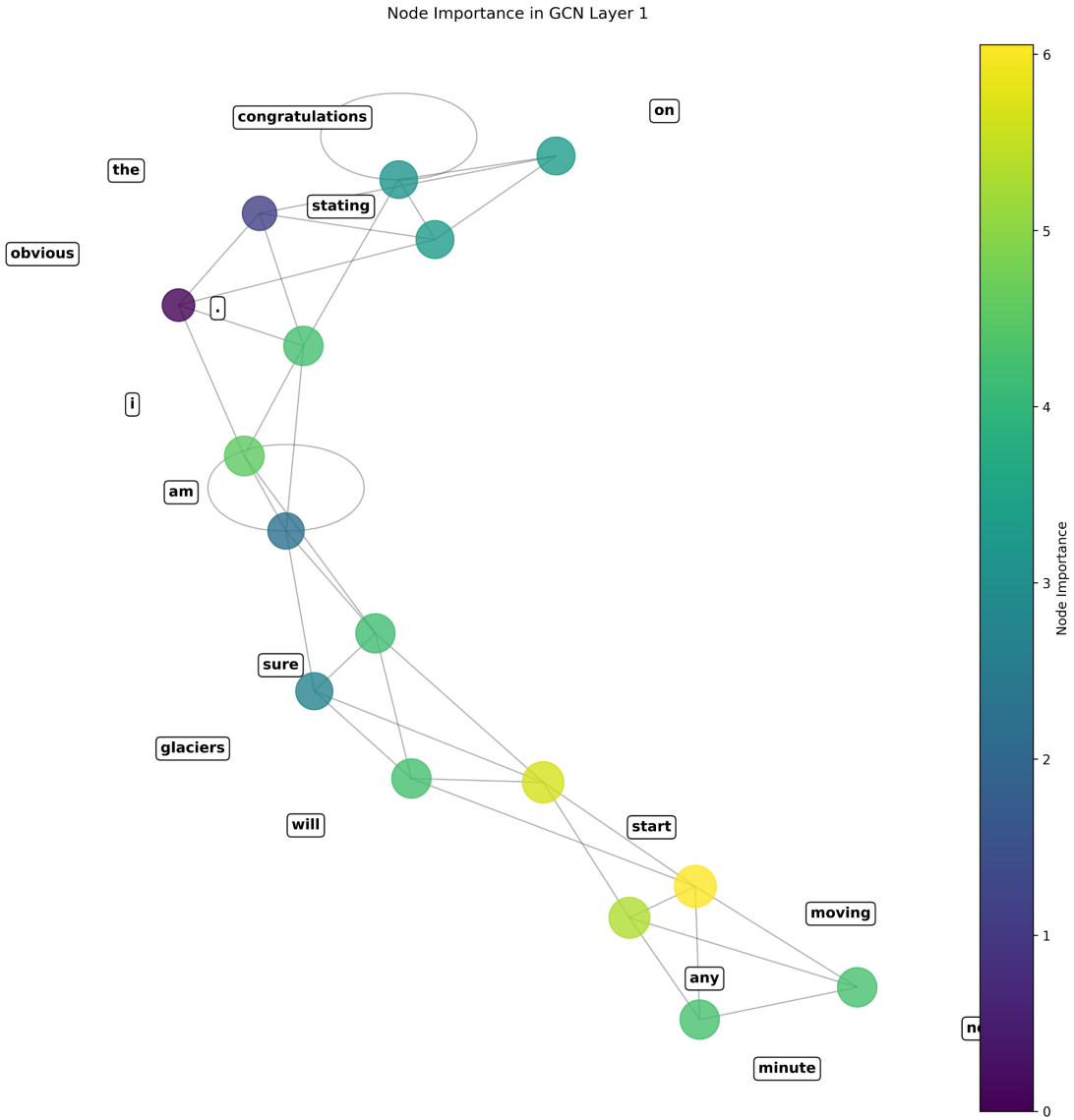
Attention Mechanism Analysis

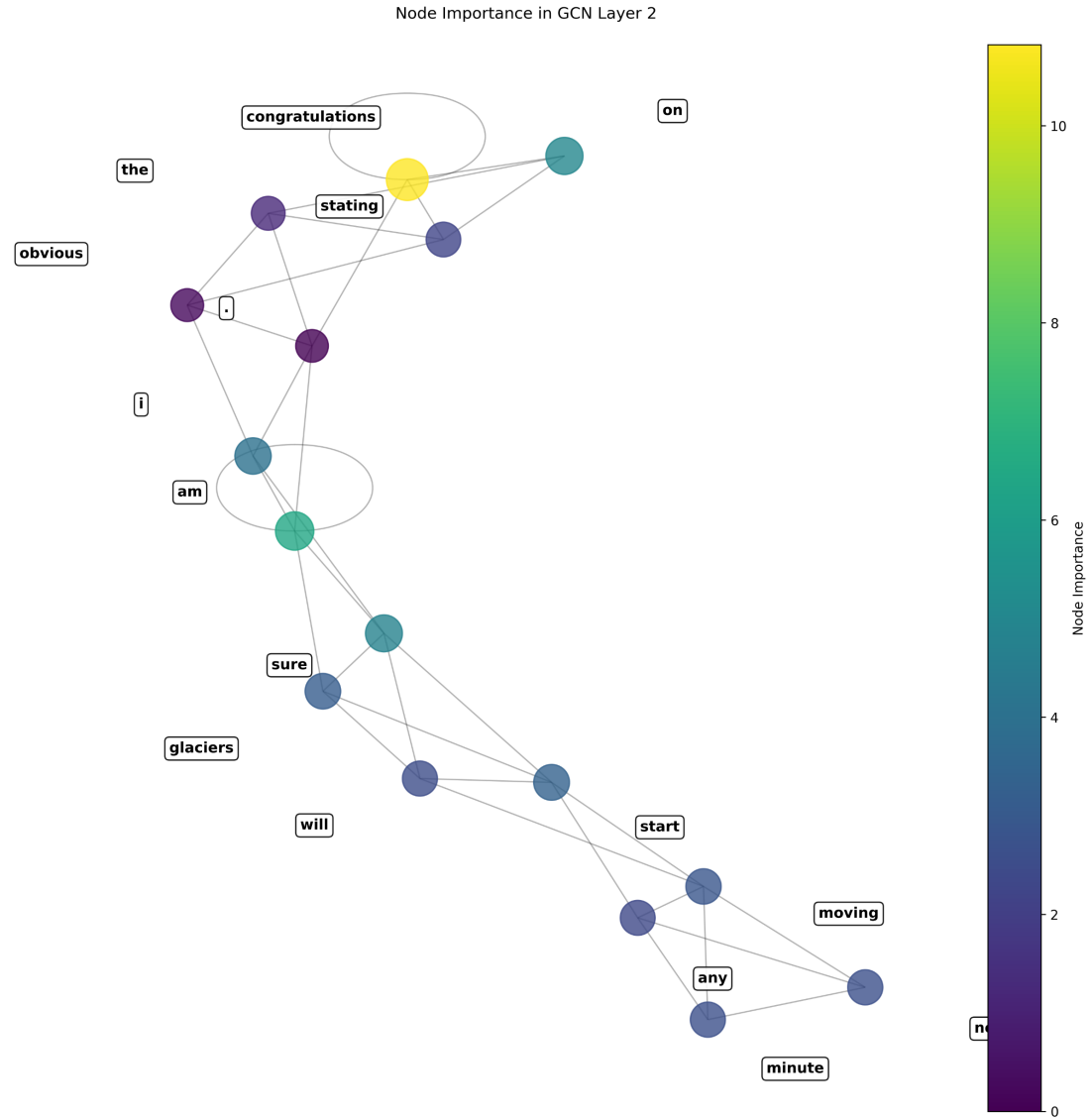


GCN Layer Evolution Analysis

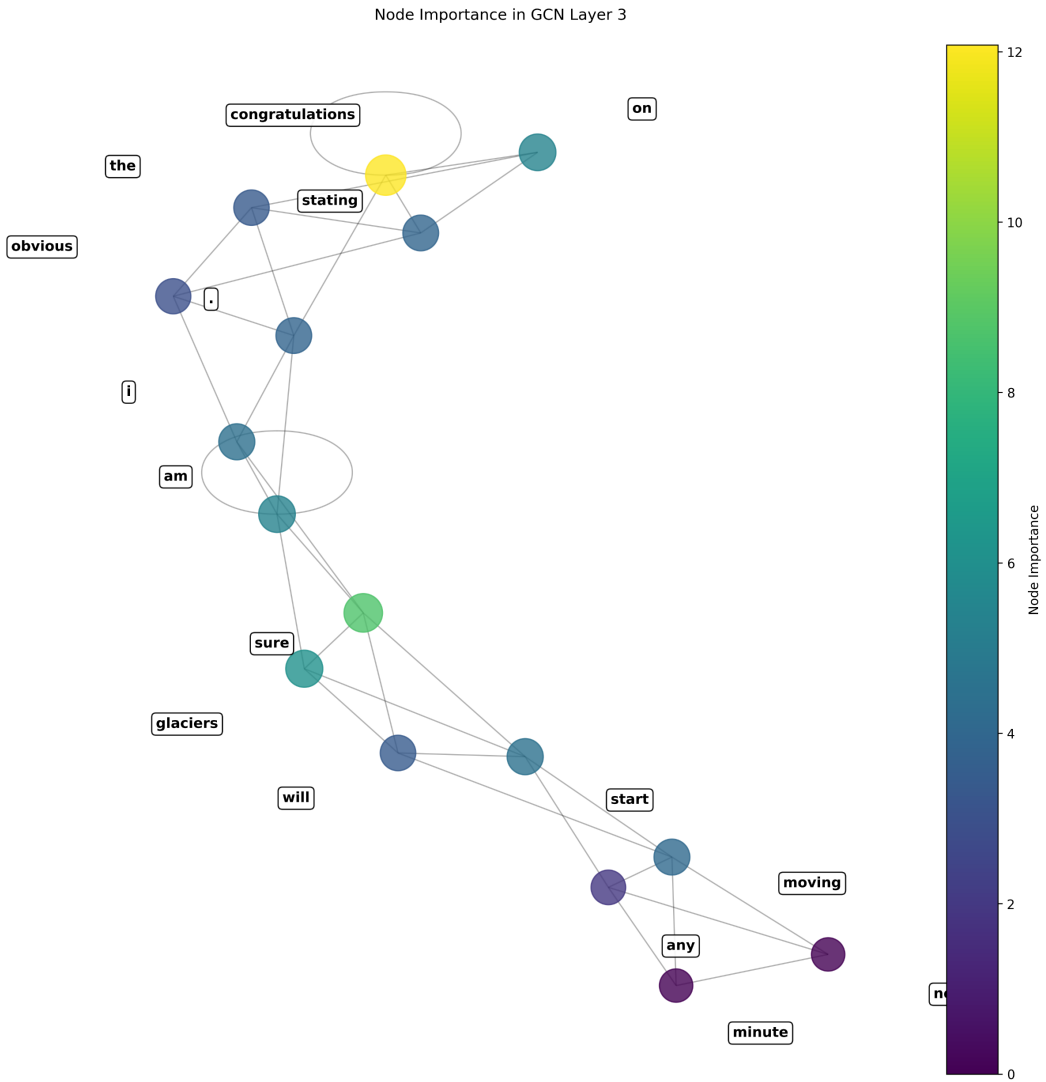


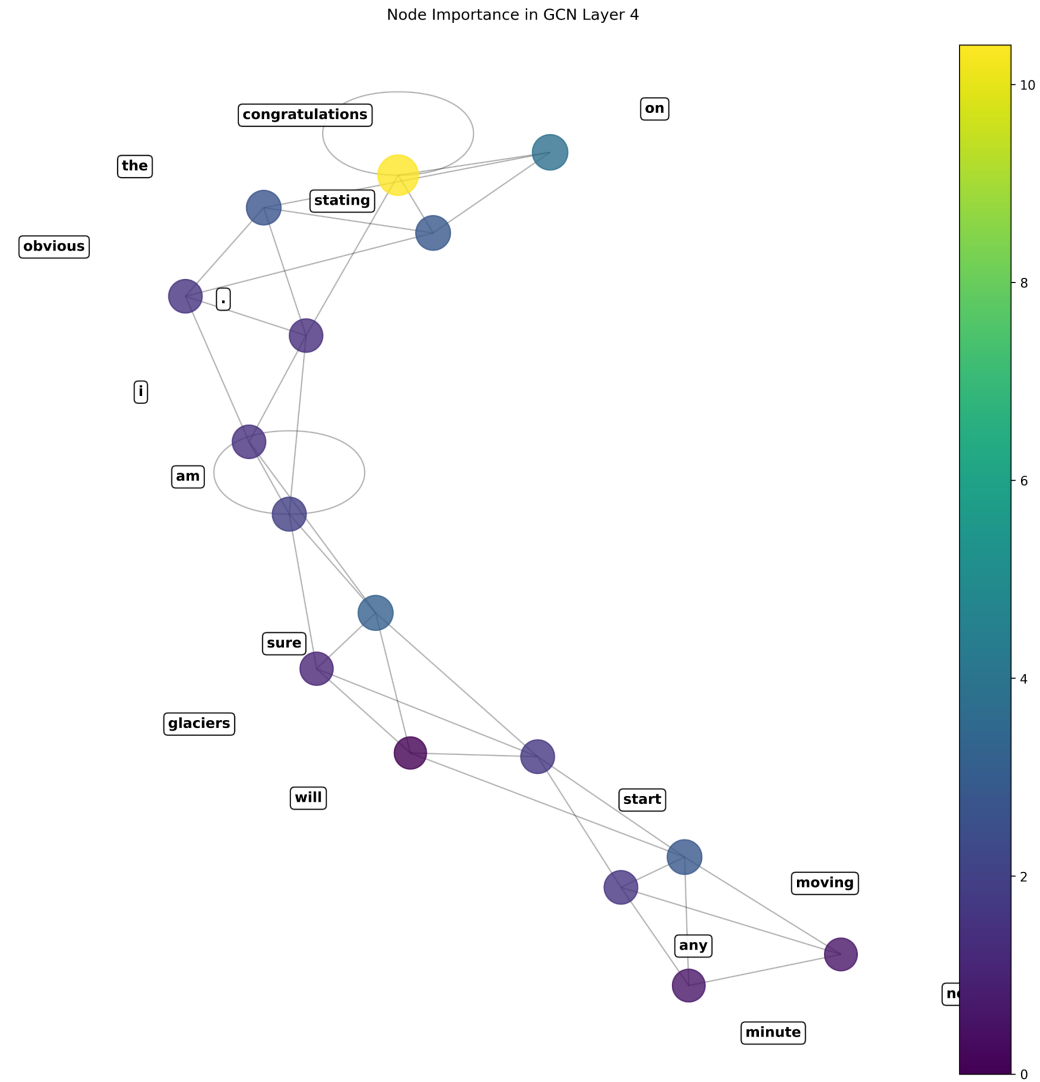
Layer-Specific GCN Analysis



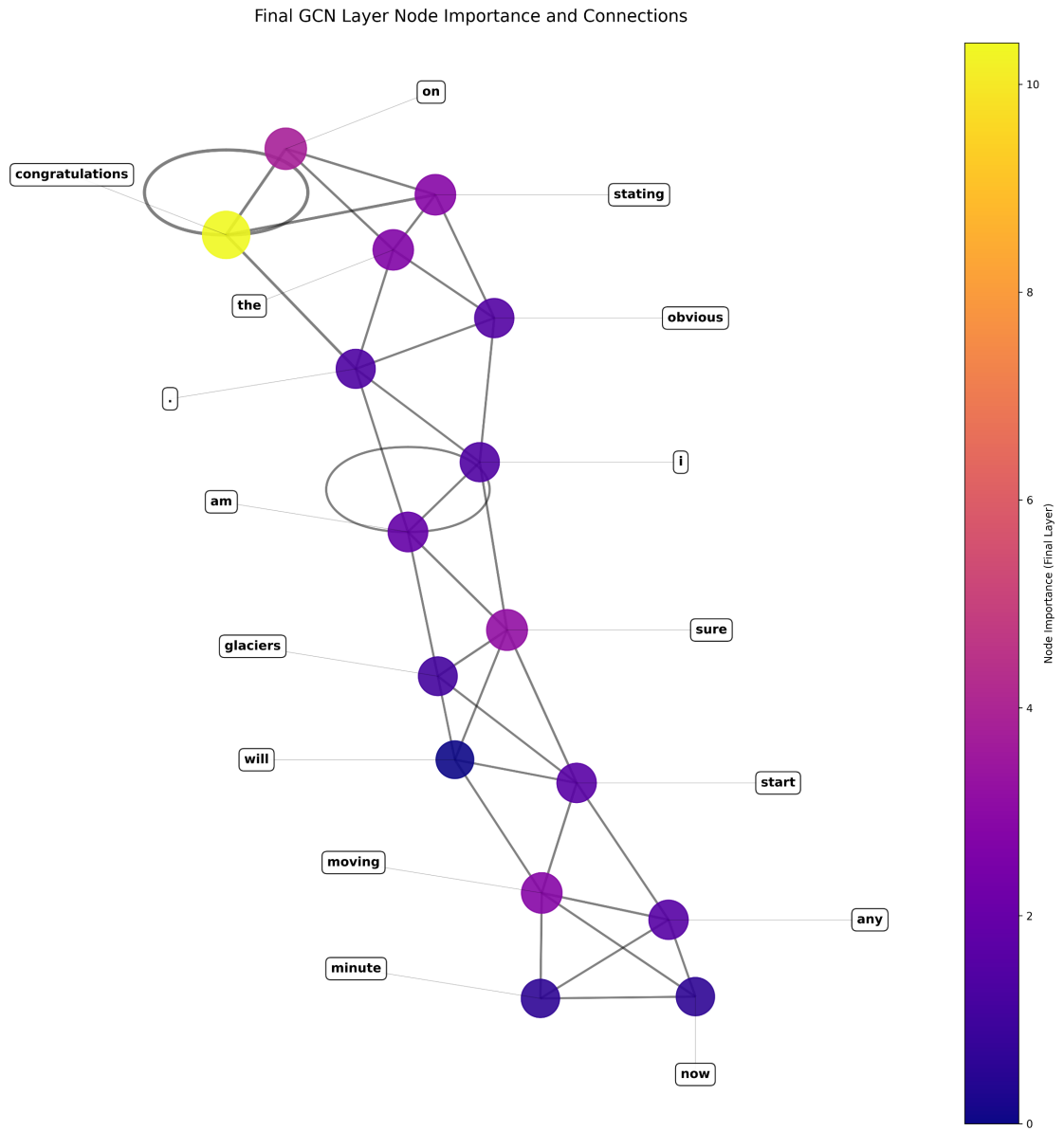


Layer-Specific GCN Analysis

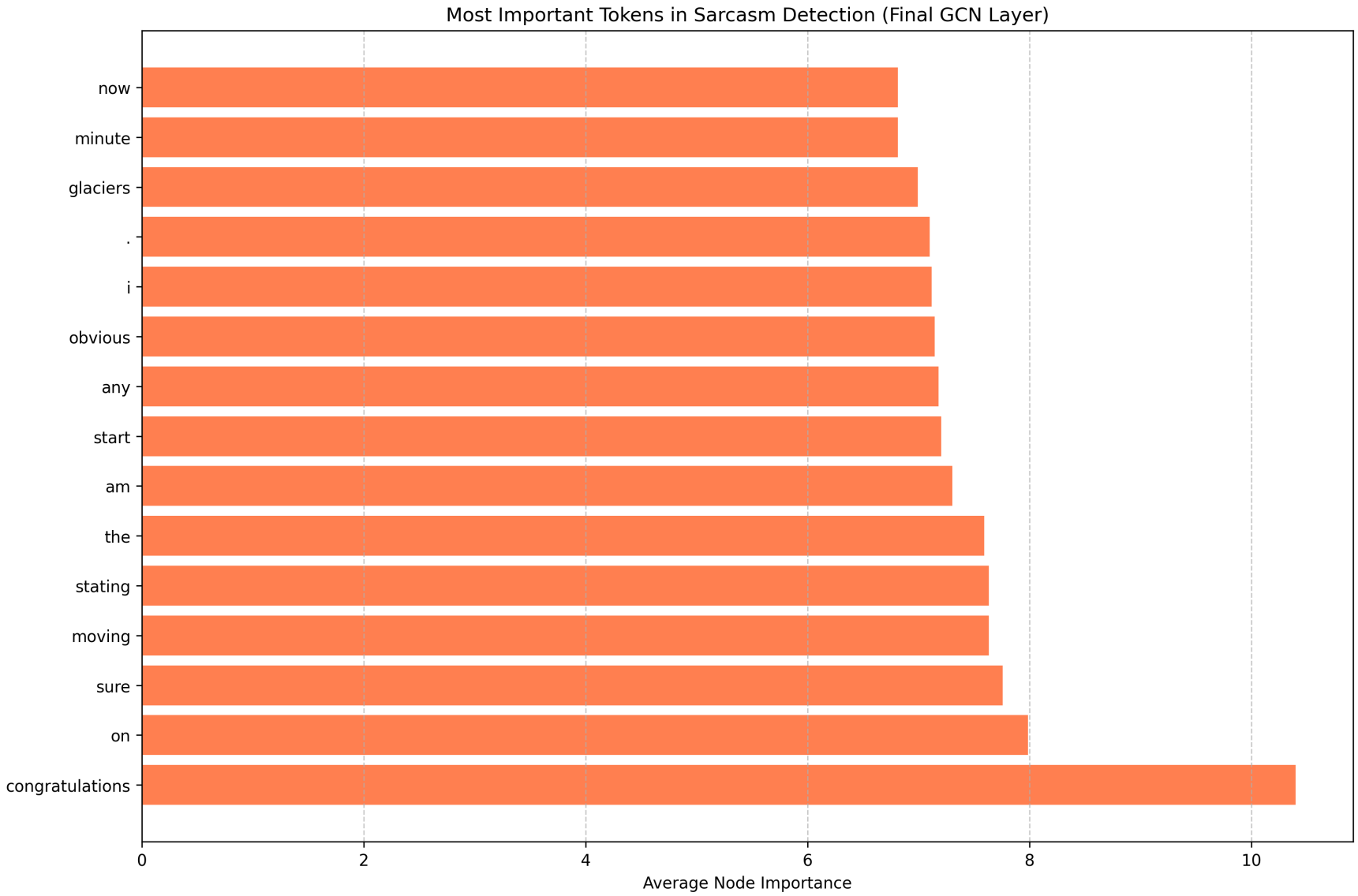




Final Layer GCN Analysis



Final Node Importance Analysis



Limitations & Challenges

- **Detection Challenges**:
  - Cultural and contextual sarcasm requiring world knowledge
  - Very subtle forms of sarcasm with minimal lexical cues
  - Domain-specific sarcasm requires specialized training
- **Rewriting Difficulties**:
  - Preserving exact semantic intent during transformation
  - Handling multi-sentence sarcastic expressions
  - Maintaining coherence in complex transformations
- **Technical Limitations**:
  - Computational demands of graph processing for long texts
  - Limited by the size of available labeled datasets
  - Challenge of evaluating rewriting quality objectively

Thank You