Multimodal Node Classification with Graph Neural Networks

Vinay R Jumani

31st March 2025

1 Introduction

This report presents a graph neural network (GNN) approach for node classification using the AMPLUS dataset. The goal is to predict entity types based on graph structure and textual features. We employ a Graph Convolutional Network (GCN) architecture to leverage both relational information and textual entity descriptions.

2 Methodology

2.1 Dataset

The AMPLUS dataset contains:

- 1,153,679 entities
- 33 relation types
- 8 classes (entity types)

2.2 Data Preprocessing

1. **Textual Feature Extraction**: Entity descriptions were analyzed to infer types using heuristic rules:

```
def infer_entity_type(text):
    if not text: return "Empty"
    elif "(" in text and ")" in text: return "Parenthetical"
    elif any(char.isdigit() for char in text): return "Numeric"
    elif len(text.split()) > 5: return "Descriptive"
    else: return "Other"
```

2. **Graph Construction**:
   - Edge index created from triplets (subject, relation, object)
   - Node features: One-hot encoding of inferred entity types

2.3 Model Architecture

GCN architecture with two convolutional layers:

```
class GCN(torch.nn.Module):
    def __init__(self):
        super(GCN, self).__init__()
        self.conv1 = GCNConv(len(unique_types), 16)
        self.conv2 = GCNConv(16, len(unique_types))

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index)
        x = F.relu(x)
        x = self.conv2(x, edge_index)
        return F.log_softmax(x, dim=1)
```

3 Results

3.1 Entity Type Distribution

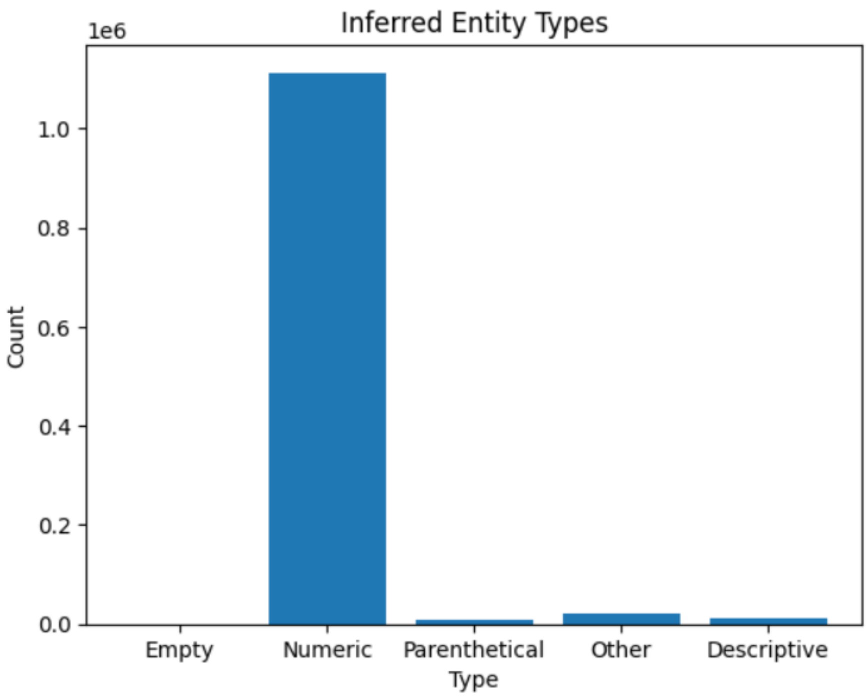

Figure 1: Distribution of inferred entity types

3.2 Training Performance

- Final training accuracy: 99.22%
- Final test accuracy: 99.18%
- Loss decreased from 0.4565 to 0.0794 over 240 epochs

3.3 Edge Type Analysis

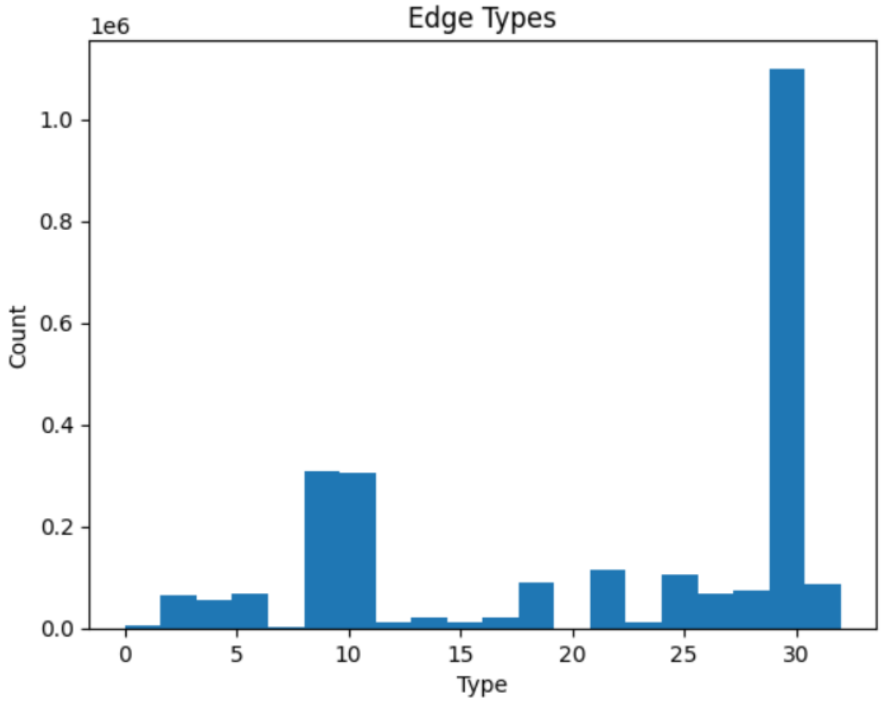

Figure 2: Distribution of relation types in the graph

4 Discussion

The model demonstrates strong performance, likely due to:

- Effective feature extraction from entity descriptions
- Proper utilization of graph structure through GCN
- Balanced train/test split (80/20)

Limitations:

- Simple heuristic-based feature engineering
- No utilization of image data available in AMPLUS
- Potential overfitting despite regularization

5 Conclusion

This work shows the effectiveness of GCNs for node classification tasks in knowledge graphs. Future work should incorporate multimodal features (text + images) and explore more sophisticated feature extraction techniques.

A Full Training Log

```
Epoch 10, Loss: 0.4565, Train Accuracy: 0.9642, Test Accuracy: 0.9644
...
Epoch 240, Loss: 0.0794, Train Accuracy: 0.9922, Test Accuracy: 0.9918
```