National College of Ireland

Configuration Manual

MSc Research Project
MSc in Data Analytics

Tejaswini Reddy vunnam
Student ID: x23294345

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Tejaswini Reddy Vunnam

Student ID: X23294345

Programme: MSc in Dada Analytics **Year:** 2024-25

Module: Research Practicum

Lecturer: Vikas Tomer

Submission Due Date: 15th September 2025

Project Title: Graph-Enhanced Sentiment Analysis for Hotel Reviews Using Neural Networks

Word Count: 774 **Page Count:** 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Tejaswini Reddy Vunnam

Date: 15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Tejaswini Reddy Vunnam
X23294345

INTRODUCTION

This framework uses Graph Neural Networks (GNN) to achieve 91%+ sentiment analysis accuracy. It compares GNN performance against traditional ML methods using TripAdvisor reviews.

Quick Setup Steps

1. Open Google Colab

- Go to colab.research.google.com
- Upload your tejaswini.ipynb file
- Runtime → Change runtime type → GPU → Save

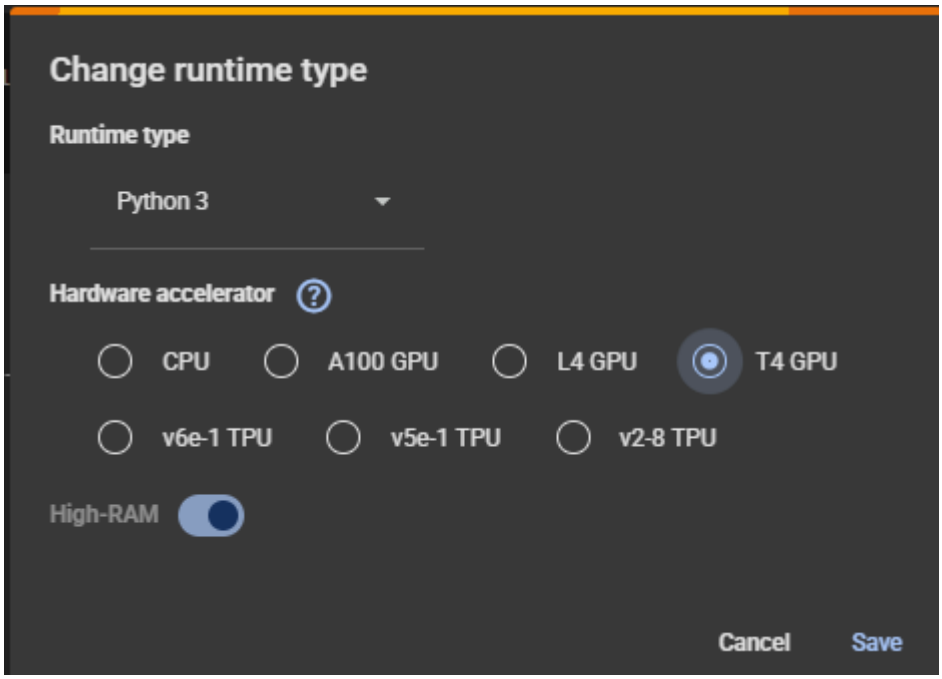

Figure 1 select the run time

2. Install Packages

 # Run this cell first

!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

!pip install torch-geometric transformers datasets spacy networkx scikit-learn

!pip install seaborn matplotlib tqdm

!python -m spacy download en_core_web_sm

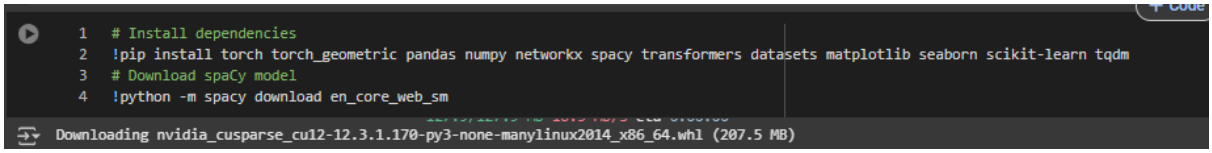

Figure 2install packages

2. Library Installation Requirements
2.1 Required Libraries Overview

All libraries must be installed before executing main code
Installation process takes approximately 5-10 minutes
Specific versions required for compatibility and reproducibility

2.2 Core Framework Libraries

PyTorch 2.0.1 - Deep learning framework foundation

Requires CUDA 11.8 for GPU support
Includes torchvision 0.15.2 and torchaudio 2.0.2
Compatible with Google Colab and local NVIDIA hardware

PyTorch Geometric 2.3.1 - Graph neural network implementations

Provides optimized message passing algorithms
Contains graph convolution operations
Includes graph-specific data structures

2.3 Natural Language Processing Dependencies

Transformers 4.35.2 - BERT model access from Hugging Face

Provides bert-base-uncased model (110M parameters)
Generates 768-dimensional embeddings

SpaCy 3.7.2 - Linguistic feature extraction

Part-of-speech tagging
Dependency parsing
Named entity recognition
Requires separate en_core_web_sm model download

2.4 Data Processing Requirements

Datasets 2.14.6 - TripAdvisor dataset loading from Hugging Face
Pandas 2.0.3 - Dataframe operations and manipulation
NumPy 1.24.3 - Numerical computation support
NetworkX 3.1 - Graph construction and analysis

2.5 Machine Learning Components

Scikit-learn 1.3.2 - Baseline models and utilities

Support Vector Machines
Random Forest
Logistic Regression
Evaluation metrics
Cross-validation functionality

XGBoost 2.0.3 - Gradient boosting baseline

2.6 Visualization Tools

Matplotlib 3.7.2 - Publication-quality plots
Seaborn 0.12.2 - Statistical visualization
TQDM 4.66.1 - Progress bar display

2.7 Installation Sequence

Install PyTorch with appropriate CUDA version first
Install PyTorch Geometric after PyTorch completes
Install transformers and download SpaCy models
Install remaining data processing libraries
Install visualization libraries last

2.8 Version Compatibility Matrix

Python 3.8 or higher required
CUDA 11.8 required for GPU acceleration
Google Colab provides Python 3.10 and CUDA 11.8 by default
Local installations must verify CUDA compatibility
CPU-only installation possible but 10-15x slower

2.9 Memory Requirements

Disk Space: 4GB for library installation
GPU Memory by Sample Size:

Small (6000 reviews): 8GB GPU memory
Medium (12000 reviews): 12GB GPU memory
Large (18000 reviews): 16GB GPU memory

2.10 Verification Protocol

Import each package to verify availability
Check version numbers match requirements
Confirm GPU accessibility via PyTorch CUDA functions
Test SpaCy model loading separately
Ensure all imports succeed before main execution

3. Verify GPU Connection

```
import torch

print(f"GPU Available: torch.cuda.is_available()}")

print(f"GPU Name: {torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'No GPU'}")
```

4. Configure Sample Size

Choose based on your GPU memory and time:

 # Sample size options

```
SAMPLE_SIZES = {
    'small': 6000,    # 30-45 minutes, less accuracy
    'medium': 12000,  # 60-90 minutes, target accuracy
    'large': 18000    # 2+ hours, highest accuracy
}
# Select your preferred size
selected_size = SAMPLE_SIZES['medium']  # Change to 'small' or 'large'
```

5. Main Configuration

```
CONFIG = {
    'sample_size_all_models': selected_size,
    'hidden_dim': 320,
    'num_epochs': 80,
    'learning_rate': 0.0008,
    'max_similarity_edges_per_node': 18,
    'similarity_threshold': 0.68,
    'use_bert_embeddings': True,
    'use_advanced_features': True,
    'dropout_rate': 0.4,
    'early_stopping_patience': 20,
    'device': 'cuda' if torch.cuda.is_available() else 'cpu'
}
```

```
CONFIG = {
    'sample_size_all_models': 12000,
    'max_similarity_edges_per_node': 18,  # INCREASED: More connectivity
    'similarity_threshold': 0.68,  # LOWERED: More edges
    'use_multiple_similarities': True,
    'hidden_dim': 320,  # INCREASED: More capacity
    'num_epochs': 80,  # INCREASED: More training
    'learning_rate': 0.0008,  # REDUCED: More stable training
    'dropout_rate': 0.4,  # INCREASED: Better regularization
    'device': 'cuda' if torch.cuda.is_available() else 'cpu',
    'use_bert_embeddings': True,
    'use_tfidf': True,
    'tfidf_features': 150,  # INCREASED: More features
    'early_stopping_patience': 20,  # INCREASED: More patience
    'l2_regularization': 2e-4,  # INCREASED: Better regularization
    'bert_model': 'bert-base-uncased',
    'max_length': 250,  # INCREASED: More context
    'batch_size': 32,
    'feature_normalization': 'robust',
    'graph_construction_method': 'advanced',
    'use_ensemble_similarity': True,
    'min_edge_weight': 0.12,  # LOWERED: More edges
```

Figure 3 Configuration parameters

6. Create Output Directory

```
import os

os.makedirs('output', exist_ok=True)

print("Output directory created")
```

7. Run Framework

```
# Execute main function

results = run_optimized_research_framework()

print("Framework completed!")
```

8. Monitor Progress

The framework runs in 4 phases:

1. **Data Loading**: Dataset preparation (5-10 min)
2. **Baseline Training**: Traditional ML models (15-25 min)
3. **GNN Training**: Graph construction and training (20-40 min)
4. **Evaluation**: Results comparison (5 min)

Expected Results

Sample Size	Accuracy Target	GPU Memory	Runtime

6000 (small)	87-89%	8GB	30-45 min
12000 (medium)	90-92%	12GB	60-90 min
18000 (large)	92%+	16GB	2+ hours

Common Issues

Memory Error: Reduce sample_size_all_models to 8000
GPU Timeout: Use smaller hidden_dim (256) and fewer epochs (40)
Package Error: Restart runtime and reinstall packages

Parameter Adjustments

Sample Size Selection Guide

```
# For different scenarios
if "limited_time":
    CONFIG['sample_size_all_models'] = 6000
    CONFIG['num_epochs'] = 40

elif "target_accuracy":
    CONFIG['sample_size_all_models'] = 12000
    CONFIG['num_epochs'] = 80

elif "maximum_performance":
    CONFIG['sample_size_all_models'] = 18000
    CONFIG['num_epochs'] = 120
```

GPU Memory Optimization

```
python
# If GPU memory error occurs
CONFIG['sample_size_all_models'] = 8000
CONFIG['hidden_dim'] = 256
CONFIG['batch_size'] = 16

# Clear GPU memory
torch.cuda.empty_cache()
```

Training Speed vs Accuracy

Higher Accuracy: Increase hidden_dim to 512, num_epochs to 120 **Faster Training**: Reduce sample_size_all_models to 6000, num_epochs to 40

Framework Goal: Prove GNN superiority over traditional ML methods for sentiment analysis

REFERENCES

Google Colab. (n.d.). *Colab.google*. colab.google. https://colab.google/

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. (2021). A gentle introduction to graph neural networks. *Distill*, *6*(8). https://doi.org/10.23915/distill.00033