National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Forename Surname
Student ID: XXX

School of Computing
National College of Ireland

Supervisor: XXX

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……

Student ID: ……

Programme: …… **Year:** ……..

Module: ……

Lecturer: ……

Submission Due Date: ……

Project Title: ……

Word Count: …… **Page Count:** ……

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: ……

Date: ……

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

Forename Surname
Student ID:

1 Environment Setup in Google Colab

The Sentiment Echo Detection System (SEDS) was implemented and executed entirely within the Google Colab environment to leverage its GPU-accelerated computation capabilities. Before running the model, the runtime must be configured to use a T4 GPU by navigating to Runtime , Change runtime type and selecting GPU as the hardware accelerator. This step is critical for efficient training, as the model incorporates a Graph Convolutional Network (GCN) alongside ConvLSTM and attention mechanisms, which require high-performance matrix operations. The implementation uses Python 3.11 and relies on libraries such as PyTorch, PyTorch Geometric, scikit-learn, pandas, NumPy, and Sentence Transformers. Missing dependencies can be installed directly in Colab cells using pip commands, allowing for an iterative and error-tolerant setup process. Such cloud-based Jupyter environments have been shown to significantly accelerate the development and testing of machine learning models by providing scalable compute resources without local hardware constraints (Bisong, 2019; Google, 2023).

2 Data Preparation and Upload

The system runs preprocessed datasets from Amazon and Yelp reviews, each coupled with precomputed MiniLM embeddings saved in. npy files. Otherwise, data and embeddings have to be uploaded straight into the current Colab runtime using google. colab. files module. When uploading the datasets they also need to be all the same length — this means that each review will have a 384-dimensional semantic vector. Mismatch in the number of textual entries and embeddings will lead to a runtime error building the graph. The uploaded files would be saved in the volatile Colab storage during a session, and are lost when the runtime is reset or disconnected. This style of embedding-based graph modeling is common in the realm of natural language processing where using precomputed vector representations minimizes computational overhead and allows for more reproducible experimentation (Reimers & Gurevych, 2019; Kipf & Welling, 2017).

3 Model Execution and Output Interpretation

After the environment is setup and data uploaded, the master execution script can be fully executed. The training pipeline start with loading the datasets, constructing a heterogeneous graph with review-to-review edges and user review user temporal edges, input them into a two-layer GCN model upgraded with node attention mechanism and ConvLSTM temporal processing. It then trains the model for a number of epochs, showing you classification metrics like precision, recall, F1-score as well confusion matrices directly in the colab output cells. There are no intermediate outputs saved to disk, everything is rendered in line ranging from the attention weight distributions to the final polarization histograms. This in-memory, display-

based approach is great for the kind of exploratory research done in Colab, where rapid iteration and visual feedback are more important than long-term storage. (Abadi et al., 2016; Fey & Lenssen, 2019).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283.

Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. Apress. https://doi.org/10.1007/978-1-4842-4470-8

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. arXiv preprint arXiv:1903.02428. https://doi.org/10.48550/arXiv.1903.02428

Google. (2023). Colaboratory FAQ. Retrieved from https://research.google.com/colaboratory/faq.html

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1609.02907

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 3982–3992. https://doi.org/10.18653/v1/D19-1410