Configuration Manual

MSc Research Project
Master of Science in Data Analytics

Vishal Vishvanath Sawant Dessai
Student ID: x23414766

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vishal Vishvanath Sawant Dessai
Student ID:	x23414766
Programme:	Master of Science in Data Analytics **Year:** 2025
Module:	Research Practicum
Lecturer:	Christian Horn
Submission Due Date:	11th December 2025
Project Title:	Toward Reliable Depression Detection in Negation-Heavy Twitter Corpora
Word Count:	465 words **Page Count: 4**

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: Vishal Vishvanath Sawant Dessai

Date: 11th December 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

Vishal Vishvanath Sawant Dessai
Student ID: x23414766

1 Overview

This configuration manual outlines the environment, tools, dataset procedures, and experimental setup required to reproduce the transformer-based depression detection system developed in this project. The system compares BERT, RoBERTa, and DeBERTa-v3 under strictly controlled conditions using the Multi-Labeled Depression English Tweet Corpus, containing weakly labelled, negation-rich Twitter posts. All experiments are conducted using stratified dataset splits, identical preprocessing pipelines, matched hyperparameters, and statistical evaluation to isolate pure architectural effects.

2 System Requirements

Hardware

- Primary environment: Google Colab
- GPU: Tesla T4 (16GB VRAM)

Software

- Python 3.12
- PyTorch ≥ 2.0 with CUDA
- HuggingFace Transformers ≥ 4.30
- scikit-learn, pandas, numpy,openpyxl
- matplotlib, seaborn
- tqdm, SciPy

```
!pip install -q transformers>=4.30.0
!pip install -q datasets
!pip install -q peft>=0.4.0
!pip install -q accelerate
!pip install -q scikit-learn
!pip install -q openpyxl
!pip install -q matplotlib seaborn
```

3 Dataset Configuration

Dataset Source

- Multi-Labeled Depression English Tweet Corpus, Harvard Dataverse
- 57,391 tweets originally → 42,723 tweets after preprocessing and deduplication

Preprocessing Rules

- HTML entities are decoded to restore original text
- URLs and user mentions are removed
- Negation cues, emojis, and hashtags are fully preserved
- Whitespace is normalised for consistency
- Duplicate tweets (after cleaning) are removed

```
URL_RE = re.compile(r"http\S+|www\.\S+")
MENTION_RE = re.compile(r"@\w+")
MULTISPACE_RE = re.compile(r"\s+")

def clean_tweet(text: str) -> str:
    if not isinstance(text, str):
        text = str(text)
    text = html.unescape(text)
    text = URL_RE.sub(" ", text)
    text = MENTION_RE.sub(" ", text)
    text = MULTISPACE_RE.sub(" ", text).strip()
    return text
```

Label Mapping

- 3-class: depressed=0, indifferent=1, happy=2
- Binary: depressed=1, non-depressed=0

```
  3-class distribution:
label_3class
0    15990
1     7127
2    19606
Name: count, dtype: int64

  Binary distribution:
label_binary
0    26733
1    15990
Name: count, dtype: int64
```

Data Splits

- 80% train, 10% validation, 10% test, stratified
- Seeds: 42, 123, 456

```
print("Configuration Summary:")
print(f"  Seeds: {Config.SEEDS}")
print(f"  Sample fraction: {Config.SAMPLE_FRAC * 100}%")
print(f"  Batch size: {Config.BATCH_SIZE}")
print(f"  Max sequence length: {Config.MAX_LENGTH}")
print(f"  Learning rate: {Config.LEARNING_RATE}")
print(f"  Max epochs: {Config.MAX_EPOCHS}")
print(f"  Mixed precision: {Config.USE_AMP}")
print(f"  Device: {Config.DEVICE}")
print(f"  Models: {list(Config.BACKBONES.keys())}")
```

```
Configuration Summary:
  Seeds: [42, 123, 456]
  Sample fraction: 100%
  Batch size: 64
  Max sequence length: 128
  Learning rate: 3e-05
  Max epochs: 30
  Mixed precision: True
  Device: cuda
  Models: ['bert', 'roberta', 'deberta']
```

4 Model & Training Configuration

Models Evaluated

- bert-base-uncased
- roberta-base
- microsoft/deberta-v3-base

All models share:

- Dropout(0.1)
- Linear classifier head (768 → num_classes)
- Hyperparameters (Identical for All Models)
- Sequence length: 128
- Batch size: 64
- Learning rate: 3e-5
- Optimizer: AdamW

- Warmup: 5%
- Epochs: up to 30 (early stopping patience=2)
- Loss: class-weighted cross-entropy

Experiment Design

- Total of 18 experimental runs
- Configuration: 3 models × 2 tasks × 3 random seeds
- Training metrics were saved for each epoch
- The best checkpoint was chosen using validation macro-F1

5 Evaluation & Reproducibility

Primary Metric

- Macro-F1,

computed on:

- Full test set
- Negation-heavy subset

Statistical Testing

- Paired bootstrap resampling
- All pairwise comparisons: $p < 0.0001$

Reproducibility Notes

- Google Colab T4 GPU for similar runtime behaviour
- Random seeds for Python, NumPy and PyTorch
- Use cached tokenised datasets to avoid variation
- Library versions unchanged for numerical stability

References

Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, 4171-4186.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

He, P., Gao, J. and Chen, W. (2023). DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. Proceedings of ICLR 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32, 8024-8035.