National College of Ireland

Configuration Manual

MSc Research Project
MSCAI

Likhitha Konasale Prakash
Student ID: 23160691

School of Computing
National College of Ireland

Supervisor: Taylou Maniganze

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Likhitha Konasale Prakash		
Student ID:	23160691		
Programme:	MSCAI	**Year:**	2024 - 2025
Module:	MSc Research Project		
Lecturer:	Taylou Maniganze		
Submission Due Date:	15-09-2025		
Project Title:	Enhancing Climate Change Stance Detection Through Advanced Synthetic Data Augumentation		
Word Count:	821	**Page Count:**	7

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: Likhitha Konasale Prakash

Date: 14-09-2025

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):	

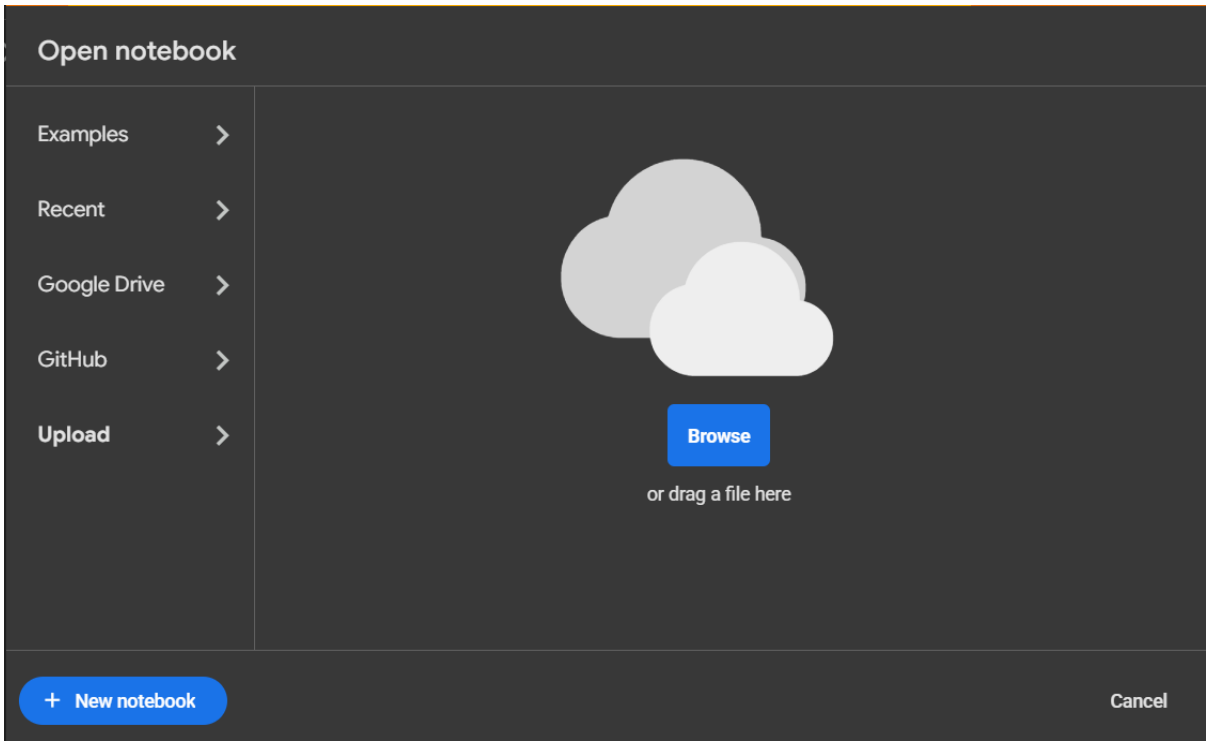
Configuration Manual

Likhitha Konasale Prakash
Student ID: 23160691

This system generates synthetic climate change tweets using AI, creating high-quality social media content with specific stances (skeptical or neutral). It uses OpenAI's GPT-4.1-mini model with advanced validation to ensure authentic-sounding posts. The system can generate hundreds of samples quickly, perfect for research, training stance detection models, or analyzing climate discourse. Everything runs in Google Colab with automatic saving to Google Drive, requiring only an OpenAI API key to get started.

Step 1: Connect to Google Colab

1. Open your web browser and go to https://colab.research.google.com
2. Sign in with your Google account
3. Click "New notebook" or File → New notebook
4. Close the new notebook tab that opens



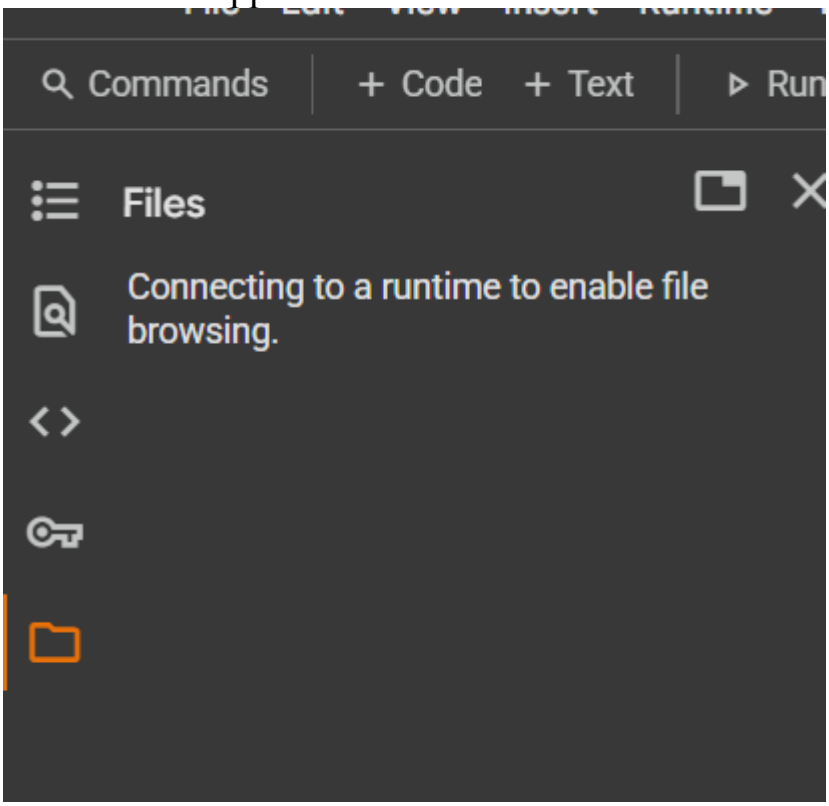
Step 2: Upload the Notebook

1. Go back to Google Colab homepage
2. Click File → Upload notebook
3. Choose "Upload" tab in the dialog box

4. Click "Browse" and select the file: Climate Stance Data.ipynb
5. Wait for upload to complete
6. The notebook will open automatically

Step 3: Upload Required Data Files

1. In the opened notebook, look at the left sidebar
2. Click the 📁 folder icon
3. Click the upload icon (paper with arrow)
4. Upload this required file:
   spicy_climate_patterns_20250719_101834.json
5. Optional files you can also upload:
   - enhanced_yake_climate_keywords.json
   - twitter_sentiment_data.csv
6. Wait for all files to appear in the file list



Step 4: Replace the API Key

1. In the notebook, scroll down or use Ctrl+F to search for:
   ParallelProcessingConfig
2. You'll find this code section around line 93:

class ParallelProcessingConfig: API_KEY = "sk-proj-YOUR_ACTUAL_OPENAI_API_KEY_HERE"

3. Replace sk-proj-YOUR_ACTUAL_OPENAI_API_KEY_HERE with your actual OpenAI API key
4. Make sure to keep the quotes around your key
5. Example after replacement:

class ParallelProcessingConfig: API_KEY = "sk-proj-abc123xyz789yourrealkeyhere"

Step 5: Run the System

1. Click Runtime → Run all (or press Ctrl+F9) to run all cells
2. Wait for all cells to execute (look for green checkmarks)
3. When execution completes, run a test:

In a new cell at the bottom, type and run:

samples = test_api(anti_count=3, neutral_count=2)

4. For production run with more samples:

samples = production_run_parallel(anti=100, neutral=100)

5. Your results will automatically save to Google Drive at: /content/drive/MyDrive/climate_generation_[TIMESTAMP]/

That's it! Your Climate Stance Data Augmentation System is now running.

Quick Verification Commands

1 Check if files uploaded correctly

!ls /content/*.json

2 Test with 5 samples

samples = test_api(anti_count=3, neutral_count=2) print(f"Generated {len(samples)} samples")

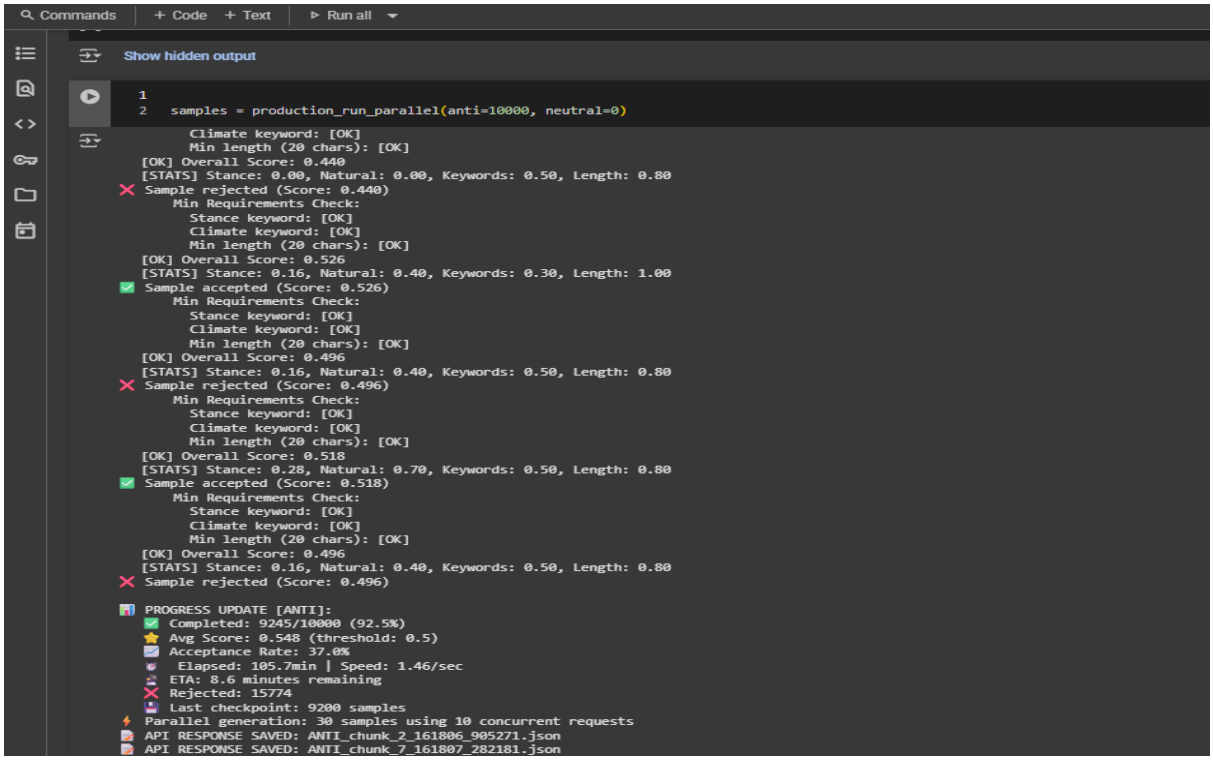
3 View a sample

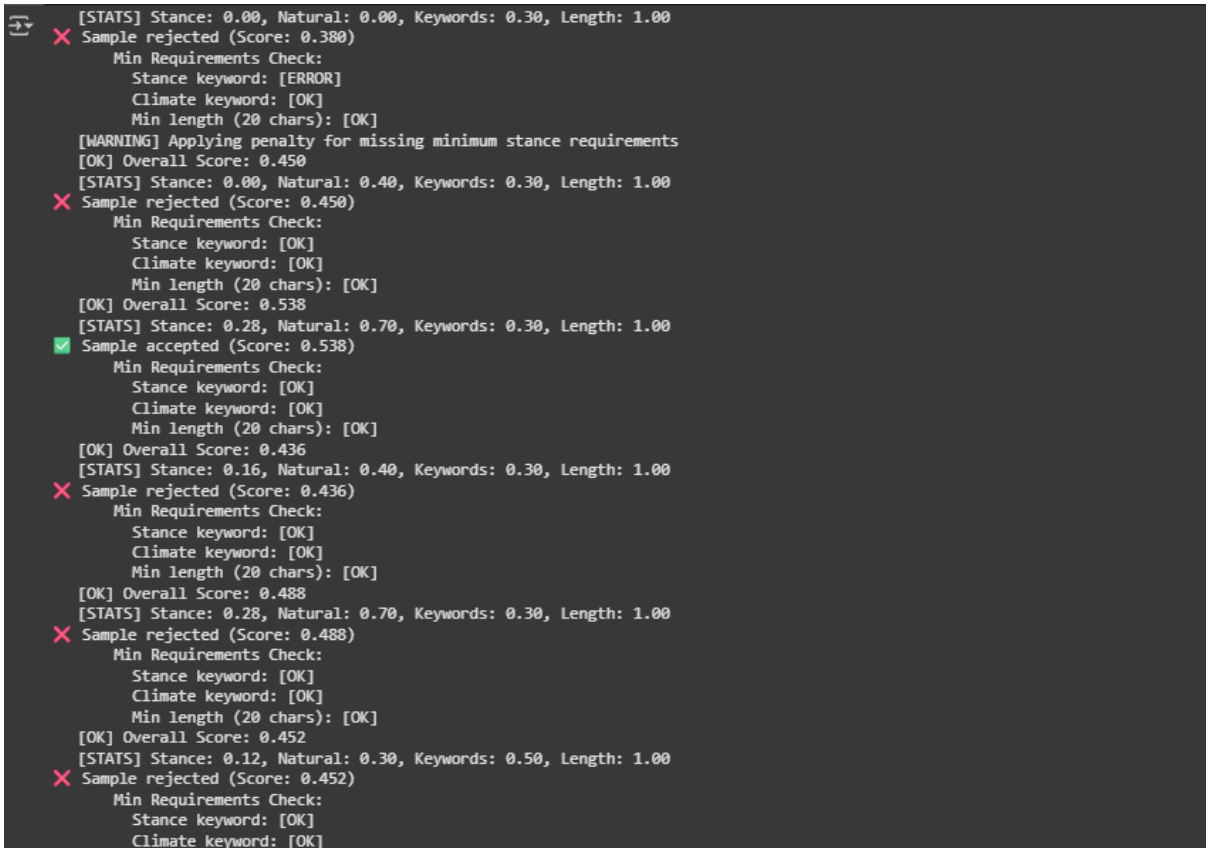
print(samples[0]['text'])

Important Reminders

- The API key must be valid and have credits
- Keep Google Colab tab open during generation
- Files auto-save to Google Drive every 50 samples
- For issues, restart runtime and try again

Outputs


Commands
+ Code
+ Text
Run all
Show hidden output
1
2 samples = production_run_parallel(anti=10000, neutral=0)
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.440
[STATS] Stance: 0.00, Natural: 0.00, Keywords: 0.50, Length: 0.80
Sample rejected (Score: 0.440)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.526
[STATS] Stance: 0.16, Natural: 0.40, Keywords: 0.30, Length: 1.00
Sample accepted (Score: 0.526)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.496
[STATS] Stance: 0.16, Natural: 0.40, Keywords: 0.50, Length: 0.80
Sample rejected (Score: 0.496)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.518
[STATS] Stance: 0.28, Natural: 0.70, Keywords: 0.50, Length: 0.80
Sample accepted (Score: 0.518)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.496
[STATS] Stance: 0.16, Natural: 0.40, Keywords: 0.50, Length: 0.80
Sample rejected (Score: 0.496)
PROGRESS UPDATE [ANTI]:
Completed: 9245/10000 (92.5%)
Avg Score: 0.548 (threshold: 0.5)
Acceptance Rate: 37.0%
Elapsed: 105.7min | Speed: 1.46/sec
ETA: 8.6 minutes remaining
Rejected: 15774
Last checkpoint: 9200 samples
Parallel generation: 30 samples using 10 concurrent requests
API RESPONSE SAVED: ANTI_chunk_2_161806_905271.json
API RESPONSE SAVED: ANTI_chunk_7_161807_282181.json


[STATS] Stance: 0.00, Natural: 0.00, Keywords: 0.30, Length: 1.00
Sample rejected (Score: 0.380)
Min Requirements Check:
Stance keyword: [ERROR]
Climate keyword: [OK]
Min length (20 chars): [OK]
[WARNING] Applying penalty for missing minimum stance requirements
[OK] Overall Score: 0.450
[STATS] Stance: 0.00, Natural: 0.40, Keywords: 0.30, Length: 1.00
Sample rejected (Score: 0.450)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.538
[STATS] Stance: 0.28, Natural: 0.70, Keywords: 0.30, Length: 1.00
Sample accepted (Score: 0.538)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.436
[STATS] Stance: 0.16, Natural: 0.40, Keywords: 0.30, Length: 1.00
Sample rejected (Score: 0.436)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.488
[STATS] Stance: 0.28, Natural: 0.70, Keywords: 0.30, Length: 1.00
Sample rejected (Score: 0.488)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]
Min length (20 chars): [OK]
[OK] Overall Score: 0.452
[STATS] Stance: 0.12, Natural: 0.30, Keywords: 0.50, Length: 1.00
Sample rejected (Score: 0.452)
Min Requirements Check:
Stance keyword: [OK]
Climate keyword: [OK]

Climate Stance Classification Modelling System

This system trains and evaluates multiple state-of-the-art transformer models (BERT, RoBERTa, DeBERTa, etc.) for climate stance classification. It implements three data strategies to leverage both original Twitter data and AI-generated augmented samples, achieving improved performance on minority classes. The system automatically trains 7 different models across 3 strategies, producing 21 trained models with comprehensive evaluation metrics.

Setup Instructions

Follow the same initial setup steps from the previous manual (Steps 1-3) for:

- Connecting to Google Colab
- Uploading the notebook file: **Likitha_Stance_Modelling.ipynb**
- Enabling GPU runtime (select T4 or A100)
- **Step 1: Update Data File Paths**

After opening the notebook, locate and verify these file paths in the code:

python

```
# Original paths in the code:
aug_anti = pd.read_csv("/content/drive/MyDrive/StanceClassificationAugmented/production_final_results_anti.csv")
aug_neutral = pd.read_csv("/content/drive/MyDrive/StanceClassificationAugmented/production_final_results_neutral.csv")
original_stance = pd.read_csv("/content/drive/MyDrive/StanceClassificationAugmented/twitter_sentiment_data.csv")
```

Required files in your Google Drive:

- /StanceClassificationAugmented/production_final_results_anti.csv (from augmentation system)
- /StanceClassificationAugmented/production_final_results_neutral.csv (from augmentation system)
- /StanceClassificationAugmented/twitter_sentiment_data.csv (original dataset)
- **Step 2: Verify Output Directories**

The system will automatically create these directories:

```
base_path = "/content/drive/MyDrive/StanceClassificationAugmented"
models_path = f"{base_path}/models"
results_path = f"{base_path}/results"
```

Step 3: Run the Complete Pipeline

1. Click **Runtime → Run all** (or press Ctrl+F9)
2. The system will execute in this order:
    - Load and combine datasets
    - Prepare three training strategies
    - Train 7 models × 3 strategies = 21 models
    - Evaluate all models
    - Save results and trained models

Training Strategies

The system implements three data strategies:

1. **Strategy 1**: Only original Twitter data
2. **Strategy 2**: Original Pro samples + Augmented Anti/Neutral samples
3. **Strategy 3**: Mixed original and augmented data

- **Models Trained**
- **BERT** (bert-base-uncased)
- **DistilBERT** (distilbert-base-uncased)
- **RoBERTa** (roberta-base)
- **DeBERTa** (microsoft/deberta-base)
- **XLNet** (xlnet-base-cased)
- **BERTweet** (vinai/bertweet-base) - Twitter-specific
- **LSTM** (Custom implementation)
- **Expected Outputs**

After completion, you'll find:

1. **Results Files**:
    - /results/all_results_detailed.csv - Complete metrics for all models
    - /results/results_summary.csv - Summary table
    - /combined_stance_dataset.csv - Unified dataset
    - /training_summary.json - Training configuration
2. **Console Output**:
    - Training progress for each model
    - Performance metrics table
    - Confusion matrices
    - Per-class precision, recall, F1 scores

- **Monitoring Progress**
- Each model takes ~5-10 minutes to train
- Total runtime: ~2-3 hours for all 21 models
- Progress bars show training status
- GPU memory usage displayed in toolbar
- **Quick Verification**

To verify successful completion:

```
# View results summary
results_df =
pd.read_csv("/content/drive/MyDrive/StanceClassificationAugmented/results
/results_summary.csv")
print(results_df)
```

Important Notes

- Keep the Colab tab open during training
- GPU runtime disconnects after 90 minutes of inactivity
- Results auto-save after each model completes
- If interrupted, you can resume from the last saved model