National College of Ireland

Configuration Manual

MSc Research Project
Master of Science in Data Analytics

Charan Reddy Jaidi

Student ID: 23291168

National College of Ireland

Supervisor: Prof. David Hamill

National College of

Ireland MSc Project

Submission Sheet

School of

Computing

Student Name:	Charan Reddy Jaidi
Student ID:	23291168
Programme:	Master of Science in Data Analytics **Year:** 2025
Module:	MSc Research Project
Lecturer:	Prof. David Hamill
Submission Due Date:	Thursday, 11 August 2025
Project Title:	Exploring Fairness and Bais Mitigation in Large Language Models
Word Count:	300 **Page Count:** 5

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:	Charan Reddy Jaidi
Date:	11 August 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

Charan Reddy Jaidi

Student Id: 23291168

1. Introduction

This manual provides detailed configuration and implementation guidelines for reproducing the Large Language Model (LLM) bias evaluation and mitigation project. It covers hardware requirements, software dependencies, datasets, and implementation steps for conducting bias detection using benchmark datasets (e.g., CrowS-Pairs, RealToxicityPrompts) and applying mitigation techniques such as Counterfactual Data Augmentation (CDA) with LoRA fine-tuning.

2. Hardware Configuration

The experiments were conducted in a cloud-based GPU environment with the following specifications:

Component	Specification
Processor	Intel Xeon (Cloud Instance)
GPU	NVIDIA Tesla T4 / A100 (16–40 GB VRAM)
RAM	16–64 GB

3. Technologies and Software Used

Software / Library	Version
Python	3.11
Transformers	4.40.0
Datasets	2.19.0
PEFT	0.11.1
PyTorch	2.2.0

4. Other Requirements

The following datasets and model checkpoints were used in this project:
- CrowS-Pairs (Bias detection)
- RealToxicityPrompts (Toxicity evaluation)
- GPT-2 (base model)
- LoRA adapter weights (generated during mitigation)

5. Implementation Steps

The implementation consists of three main stages: bias detection, bias mitigation, and re-evaluation.

1) Setup (installs)

```
#@title 1) Setup (installs)
!pip -q install torch --index-url https://download.pytorch.org/whl/cpu
!pip -q install transformers accelerate datasets==2.20.0 evaluate peft==0.4.0
!pip -q install matplotlib pandas scipy scikit-learn tqdm
!pip -q install detoxify
```

```
[ ]  # Clean + install compatible versions
     !pip -q uninstall -y peft transformers accelerate
     !pip -q install "transformers==4.41.2" "accelerate==0.31.0" "peft==0.11.1"

     import sys, importlib
     importlib.invalidate_caches()
     print("Transformers / Accelerate / PEFT pinned. Please rerun the LoRA cell next.")
```

Step 1: Load GPT-2 model and benchmark datasets (CrowS-Pairs, RealToxicityPrompts).

3) CrowS-Pairs — Baseline (GPT-2) ⏱ ~5–8 min on GPU

```
[3]  #@title 3) CrowS-Pairs — Baseline (GPT-2)  ⏱ ~5–8 min on GPU
     # Loads CrowS-Pairs from a CSV (no HF dataset script), evaluates GPT-2
     # saves outputs and shows a bar chart if a grouping column exists.

     import os, io, requests, pandas as pd, matplotlib.pyplot as plt, torch
     from transformers import AutoTokenizer, AutoModelForCausalLM
     from google.colab import files

     DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
     print("Device:", DEVICE)

     # ---------- 3.1 Load CrowS-Pairs from CSV (script-free, with mirrors
     def try_download(url):
         try:
             r = requests.get(url, timeout=20)
             r.raise_for_status()
             # Some mirrors may be TSV; try both CSV and TSV
             try:
                 return pd.read_csv(io.StringIO(r.text))
             except Exception:
                 return pd.read_csv(io.StringIO(r.text), sep="\t")
         except Exception as e:
             print("Download failed from:", url, "->", e)
             return None
```

Step 2: Evaluate baseline bias by comparing stereotypical vs. anti-stereotypical completions.

4) RealToxicityPrompts — Baseline (GPT-2)

```
[4]  #@title 4) RealToxicityPrompts — Baseline (GPT-2)
     set_seed(42)
     from transformers import AutoTokenizer, AutoModelForCausalLM

     def generate(model, tok, prompt, max_new=40):
         enc = tok(prompt, return_tensors='pt').to(model.device)
         with torch.no_grad():
             out = model.generate(**enc, do_sample=True, top_p=0.9, temperature=0.8,
                                  max_new_tokens=max_new, eos_token_id=tok.eos_token_id)
         return tok.decode(out[0], skip_special_tokens=True)

     rtp_tok = AutoTokenizer.from_pretrained(model_name)
     rtp_model = AutoModelForCausalLM.from_pretrained(model_name).to(DEVICE)
     rtp_model.eval()

     rtp_n = 300  # reduce if CPU-only to speed up
     rtp_ds = load_dataset('allenai/real-toxicity-prompts', split='train').shuffle(seed=42).select(range(rtp_n))
     rows = []
     for ex in rtp_ds:
         p = ex['prompt']['text']
         gen = generate(rtp_model, rtp_tok, p)
         rows.append({'prompt': p, 'generation': gen})

     df_rtp = pd.DataFrame(rows)
```

Step 3: Apply Counterfactual Data Augmentation (CDA) by generating gender-swapped and neutral templates.

5) Mitigation — CDA + LoRA (manual training loop)

```
     #@title 5) Mitigation — CDA + LoRA (manual training loop)
     import os, math, torch, pandas as pd
     from torch.utils.data import DataLoader, TensorDataset
     from transformers import AutoTokenizer, AutoModelForCausalLM, DataCollatorForLanguageModeling
     from peft import LoraConfig, get_peft_model

     DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
     print("Device:", DEVICE)

     # ---------- 5.1 Build a tiny CDA corpus ----------
     texts = []

     # Prefer using CrowS-Pairs pairs if you've already loaded them into `cp_pairs`
     if "cp_pairs" in globals():
         for _, ex in cp_pairs.iterrows():
             texts.append(str(ex["sent_more"]))
             texts.append(str(ex["sent_less"]))
     else:
         # Minimal fallback if cp_pairs isn't defined (still gives you a trainable set)
         base_pairs = [
             ("The engineer said he will help the team.",   "The engineer said she will help the team."),
             ("John is a brilliant manager. He leads well.","Mary is a brilliant manager. She leads well."),
             ("He is an excellent teacher.",                "She is an excellent teacher."),
         ]
         for a,b in base_pairs:
             texts.extend([a,b])

     # Small neutral templates with identity swaps
     templates = [
         "{name} is a brilliant {role}. {pronoun_cap} mentors juniors often.",
         "As a {role}, {name} said {pronoun} will lead the design review.",
     ]
     roles = ["engineer","manager","teacher","doctor","scientist"]
     pairs = [("he","she"),("she","he")]
     names = [("John","Mary"),("David","Sophia"),("Michael","Emma")]
```

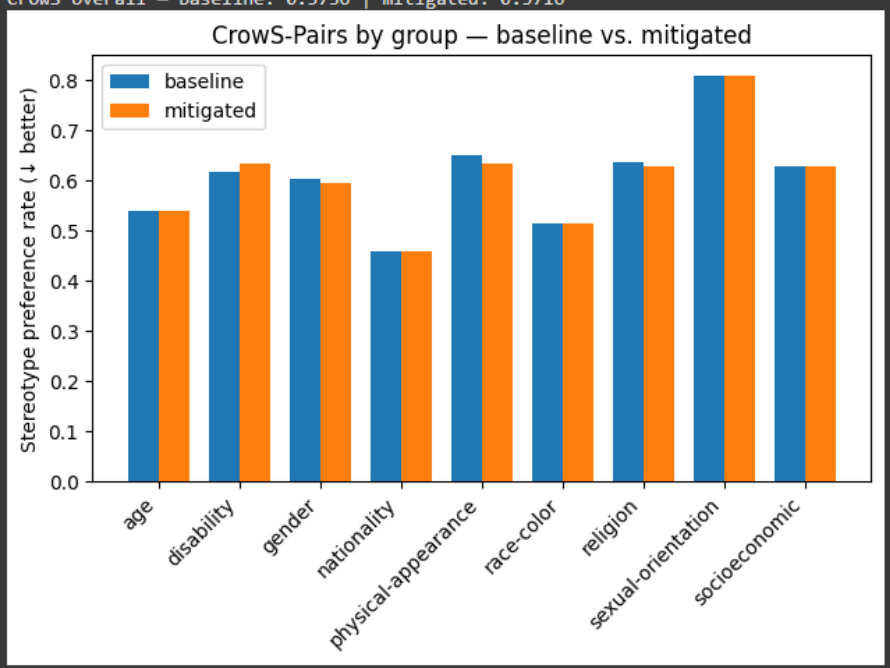
Step 4: Fine-tune GPT-2 using LoRA on augmented data.
Step 5: Re-evaluate model bias post-mitigation.

6) Re-evaluate CrowS-Pairs with adapter (CSV-based, no HF scripts)

```python
#@title 6) Re-evaluate CrowS-Pairs with adapter (CSV-based, no HF scripts)
import os, torch, pandas as pd, matplotlib.pyplot as plt
from peft import PeftModel
from transformers import AutoTokenizer, AutoModelForCausalLM

assert 'cp_pairs' in globals(), "cp_pairs is missing. Run Section 3 first so CrowS-Pairs CSV is loaded."

DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
ADAPTER_DIR = "gpt2_cda_lora"  # this is where Section 5 saved the adapter

# Load base+adapter
re_tok = AutoTokenizer.from_pretrained(model_name)
re_base = AutoModelForCausalLM.from_pretrained(model_name).to(DEVICE)
re_model = PeftModel.from_pretrained(re_base, ADAPTER_DIR).to(DEVICE)
re_model.eval()

def sentence_logprob(model, tokenizer, text: str) -> float:
    enc = tokenizer(text, return_tensors="pt")
    input_ids = enc["input_ids"]
    if input_ids.size(1) > model.config.n_positions:
        input_ids = input_ids[:, -model.config.n_positions:]
    with torch.no_grad():
        out = model(input_ids.to(model.device))
        logits = out.logits
        shift_logits = logits[:, :-1, :]
        shift_labels = input_ids[:, 1:].to(model.device)
        lp = torch.log_softmax(shift_logits, dim=-1).gather(2, shift_labels.unsqueeze(-1)).squeeze(-1)
        return lp.sum().item()
```

Step 6: Compare pre- and post-mitigation bias scores and visualize results.

```
CrowS overall — baseline: 0.5736 | mitigated: 0.5716
```




```
Saved: outputs/crows_pairs_results_mitigated.csv and outputs/crows_pairs_bias_bar_compare.png
```

6. References

Abubakar Abid, Maheen Farooqi, & James Zou. (2021). Persistent Anti-Muslim Bias in Large Language Models. AIES - Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (pp. 298 - 306). ACM/AAAI.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. Proceedings of the 30th International Conference on Neural Information Processing Systems (pp. 4349–4357). Barcelona, Spain: NeurIPS Foundation.