National College of Ireland

Configuration Manual

MSc Research Project
Master of Science in Data Analytics Information

Gaurav Sadanand Satam
Student ID: 23269898

School of Computing
National College of Ireland

Supervisor: Cristina Hava Muntean

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Gaurav Sadanand Satam		
Student ID:	23269898		
Programme:	M.Sc. Data Analytics	**Year:**	2025-2026
Module:	Research Practicum		
Supervisor:	Cristina Hava Muntean		
Submission Due Date:	11-12-2025		
Project Title:	Configuration Manual		
Word Count:	**11 Pages**	**1498 Count words**	

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: Gaurav Sadanand Satam

Date: 10/12/2025

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

Gaurav Sadanand Satam
Student ID: 23269898

1 Purpose

This Configuration Manual records the technical landscape, requirements, sources of data and the procedures to execute to replicate the analyses that have been conducted in the complementing MSc research project on Reddit discussions about ChatGPT. It is concerned with Python notebook structure and functionality, which load, preprocess, and analyse data, sentiment analysis, ethical position classification, quantifying creativity, and social network analysis. The manual is aimed at supervisors, examiners, and other future researchers who would like to reproduce or develop the work.

2 Computing Environment and Requirement

2.1 Hardware requirements

A 64-bit laptop or desktop computer that has:

- 16 GB RAM (32 GB is suggested to be faster at embedding and zero-shot).
- Multi-core processor: it should have access to a graphics card (NVIDIA, with CUDA), but it is optional.
- A minimum of 50 GB free disk to use in storing the raw dataset, intermediate CSV files, and model caches.
- Since the transformer and Sentence-BERT models are large (hundreds of megabytes to multiple gigabytes), this matters with regard to having enough RAM and disk space in order to run them stably.

2.2 Operating System

The notebook was created and tested with Windows 10/11 64-bit. The code is operating system independent and must be able to execute on Linux or even on the macOS should Python 3 and the mentioned libraries be installed properly.

2.3 Software Requirements

- Python Version: Python 3.10 or later version. In my case, I've used Python 3.13
- Successful installation of Jupyter notebook in order to interactively run the python code.
- Pip or Conda is required as package manager with relevant package versions.

3 Python Libraries and Versions

Core data and plotting:

- Pandas: used for data frame manipulation and loading csv files.
- Numpy: numerical operations.
- Matplotlib: base plotting.
- Seaborn: statistical visualisation.

NLP and transformers:

- Transformers: Hugging face pipelines in sentiment and zero-shot classification.
- sentence-transformers: Sentence-BERT embeddings (all-MiniLM-L6-v2 model).
- Nltk: lexical diversity token.

Machine learning utilities:

- Scikit-learn: From various utilities like sklearn.metrics.pairwise, 'cosine-similarity' was used.

Improvement and infrastructure:

- Tqdm: Progress bar for batch processing in order to indicate computational progress.

Network Analysis:

- NetworkX: centrality, graph construction and community.

For example, installation of required libraries should be done as demonstrated in the image below:

```
import pandas as pd
from transformers import pipeline
from tqdm import tqdm
import nltk
nltk.download('punkt')
```

Note: User needs to make sure that they download the necessary NLTK tokenizers as well.

4 Dataset Details

4.1 Source of Dataset:

- Name: "filtered_student_chatgpt_reddit.csv"
- Author: Kaggle - How Students Use ChatGPT Filtered Reddit Dataset by Muhammad Faizan Khandeshmukh. (Link)
- Records: 448,020 Reddit posts and comments that were filtered based on educational and ChatGPT-related content.

The CSV file can be downloaded easily from the Kaggle website using the provided link and needs to be stored in the working directory. In the image shown below, the python code in the Jupyter notebook assumes a relative path:

```
# Adjust the path as needed for your environment
df = pd.read_csv('filtered_student_chatgpt_reddit.csv')

print(f"Dataset shape: {df.shape}")
df.head()
```

```
Dataset shape: (448020, 10)
```

	subreddit	body	controversiality	score	Unnamed: 0	comment_id	comment_parent_id	comment_body	Topic	Tag
0	gaming	it's pretty well known and it was a paid produ...	0.0	3	NaN	NaN	NaN	NaN	NaN	NaN
1	relationship_advice	i would actually just like to say that there a...	0.0	0	NaN	NaN	NaN	NaN	NaN	NaN
2	news	the character 令 does bear a meaning similar to...	0.0	8	NaN	NaN	NaN	NaN	NaN	NaN
3	SquaredCircle	wrong. young footballers to this day help arou...	0.0	25	NaN	NaN	NaN	NaN	NaN	NaN
4	news	thanks for the interesting response, this shed...	0.0	0	NaN	NaN	NaN	NaN	NaN	NaN

4.2 Key Features:

The raw CSV to be used in the notebook will have the following columns:

- subreddit: Reddit community/forum of each post/comment.
- body: Reddit post content of the main text.
- controversiality: This is a yes/no indicator that shows whether the post/comment is controversial.
- score: Reddit score (upvotes-downvotes)
- Unnamed: 0: Row index (not important when analysing)
- comment_id: An identifier of a comment that is unique.
- comment_parent_id: This is an identifier, which connects a comment with its parent (this allows network/thread analysis).
- command: comment, reply, etc.: Text of the comment/reply.
- topic: Topic assignment (could be pre-existing or could be topic modelling generated)
- tag: Extra label (e.g., sentiment, ethical/creative indicators)

In case of a rename of fields in future versions of the dataset, users have to modify the column names in the loading and preprocessing steps.

5 Notebook Form and Implementation Process

The notebook is arranged into rational parts that ought to be carried out in a top-to-bottom fashion.

5.1 Imports and dataset loading:

Section 2 cells of the python notebook load all the required packages and aids to load the raw CSV into a pandas data frame named as 'df'. Note that:

- 'pd.set_option('display.max_columns', None)' is a command used in inspecting data frames to display all the columns.
- 'warnings.filterwarnings('ignore')' helps avoid indication of warnings that are not that important ensuring that the output is more indicative and readable.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
```

5.2 Preliminary investigation and preparation:

Section 3 performs:

- List of columns, 'df.info()', 'df.describe(include='all').T' to investigate data types, missing values and descriptive statistics to learn about distributions, such as the distributions of scores, widespread etc.

```
print("Dataset Columns:", df.columns.tolist())
print("\nBasic Info:")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
print(df.describe(include='all').T)
```

[6]

```
Dataset Columns: ['subreddit', 'body', 'controversiality', 'score', 'unnamed: 0', 'comment_id', 'comment_parent_id', 'comment_body', 'topic', 'tag']

Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 448020 entries, 0 to 448019
Data columns (total 10 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   subreddit          12202 non-null   object
 1   body               448020 non-null  object
 2   controversiality   12202 non-null   float64
 3   score              448020 non-null  int64
 4   unnamed: 0         0 non-null       float64
 5   comment_id         0 non-null       float64
 6   comment_parent_id  0 non-null       float64
 7   comment_body       0 non-null       float64
 8   topic              435818 non-null  float64
 9   tag                435818 non-null  object
dtypes: float64(6), int64(1), object(3)
memory usage: 34.2+ MB
None

Missing Values:
 subreddit            435818
body                      0
```

- Column names should be standardised to lower case and devoid of spaces followed with removal of duplicates and converting columns to numeric where relevant as shown below:

3.2 Feature Engineering & Transformation

```
# Standardizing column names for easy access
df.columns = [col.lower().strip() for col in df.columns]

# Checking and removing duplicates
print(f"Duplicated rows: {df.duplicated().sum()}")
df = df.drop_duplicates()

# Converting columns to numeric where relevant
df['score'] = pd.to_numeric(df['score'], errors='coerce').fillna(0).astype(int)
df['controversiality'] = pd.to_numeric(df['controversiality'], errors='coerce').fillna(0).astype(int)
```

[7]

```
Duplicated rows: 34042
```

5.3 Handling missing values:

A heatmap of missing values is created using ‘sns.heatmap(df.isnull(), cbar=False)’. To make analysis practical, the text and label fields (topic, tag, body, comment_body) are populated with the strings of "Unknown" and the structural columns that are not used are eventually dropped.

3.3 Handling Missing Values

```
# Visualizing missing values
sns.heatmap(df.isnull(), cbar=False)
plt.title('Missing Values Heatmap')
plt.show()

# Fill NAs in key textual/categorical columns with 'Unknown'
for col in ['topic', 'tag', 'body', 'comment_body']:
    if col in df.columns:
        df[col] = df[col].fillna('Unknown')
```

[8]

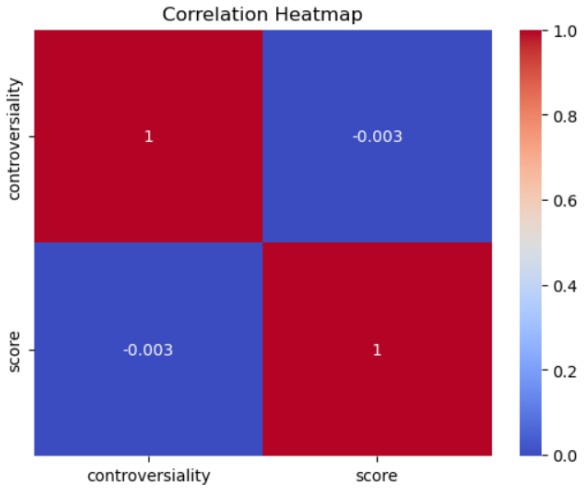
6 Exploratory Data Analysis

Distributions and filtering guide require EDA to comprehend. There are four principal visualisations set by the notebook:

- Correlation heatmap of controversiality and score.

```
numeric_cols = ['controversiality', 'score']
sns.heatmap(df[numeric_cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

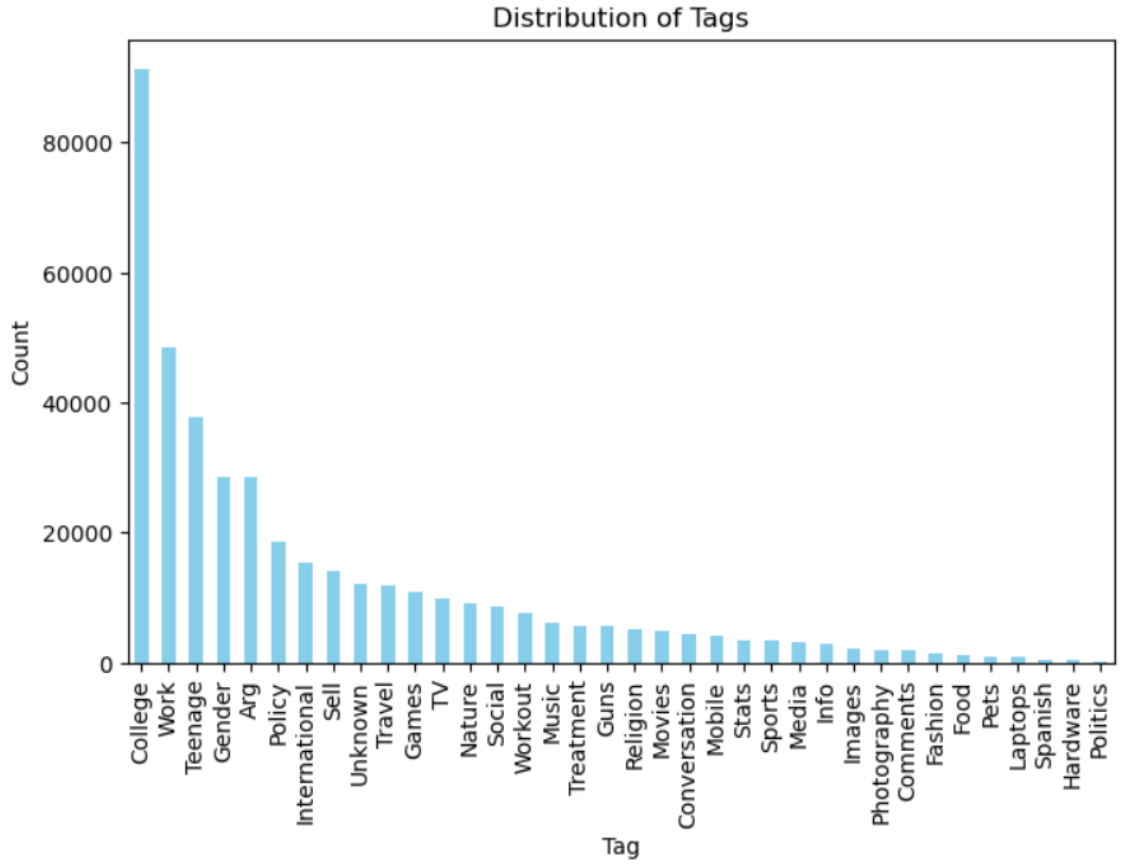
[9]

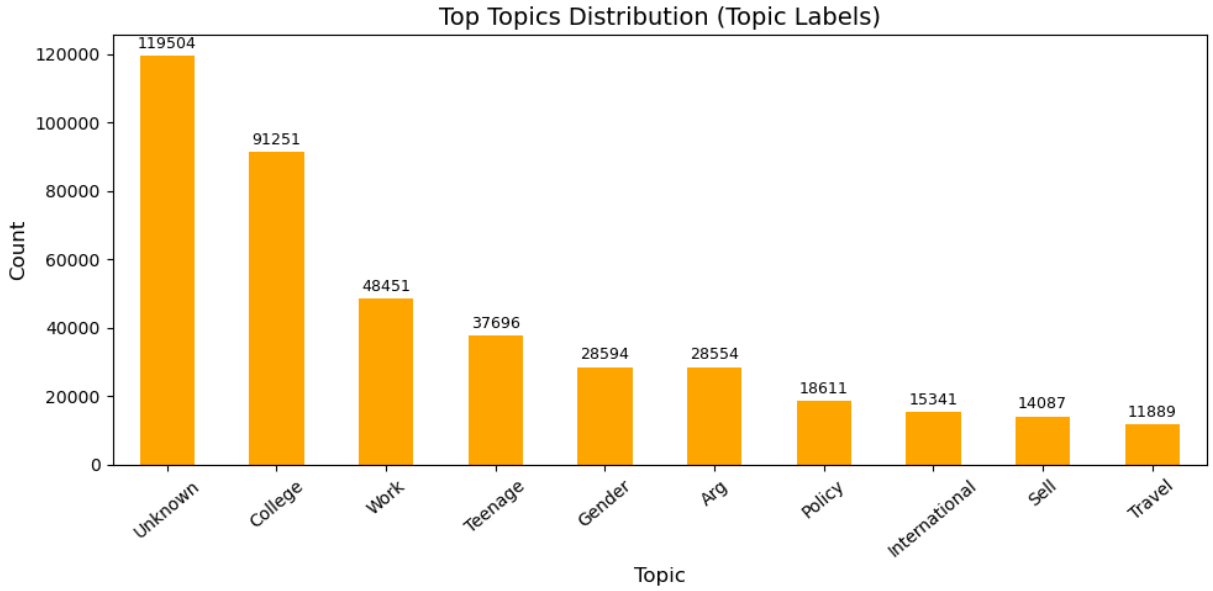


- Distribution of sub reddit ‘tags’ using ‘df['tag'].value_counts().plot(kind='bar', color='skyblue')’ and ‘topics’ using ‘df['topic'].value_counts().head(10).plot(kind='bar', color='orange')’ with the help of histogram.

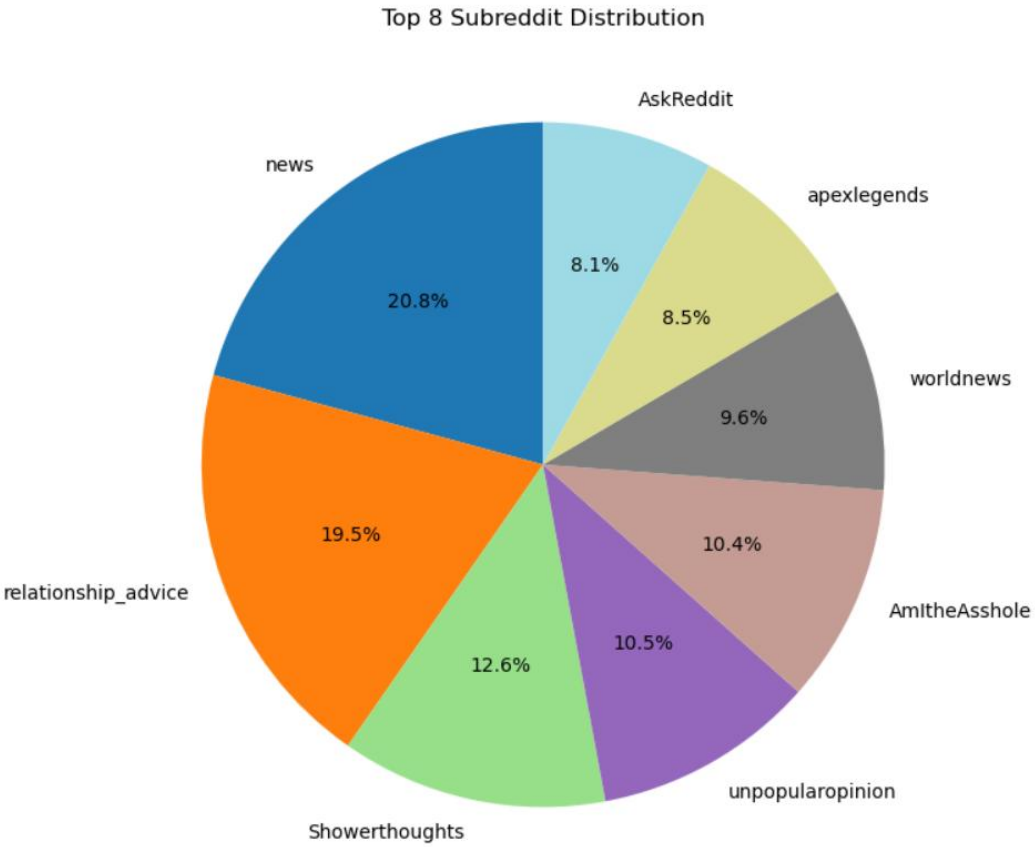
```
if 'tag' in df.columns:
    plt.figure(figsize=(8,5))
    df['tag'].value_counts().plot(kind='bar', color='skyblue')
    plt.title('Distribution of Tags')
    plt.xlabel('Tag')
    plt.ylabel('Count')
    plt.show()

if 'topic' in df.columns:
    plt.figure(figsize=(10,5))
    df['topic'].value_counts().head(10).plot(kind='bar', color='orange')
    plt.title('Top 10 Topics Distribution')
    plt.xlabel('Topic')
    plt.ylabel('Count')
    plt.show()
```

[10]


Distribution of Tags
Count
80000
60000
40000
20000
0
College
Work
Teenage
Gender
Arg
Policy
International
Sell
Unknown
Travel
Games
TV
Nature
Social
Workout
Music
Treatment
Guns
Religion
Movies
Conversation
Mobile
Stats
Sports
Media
Info
Images
Photography
Comments
Fashion
Food
Pets
Laptops
Spanish
Hardware
Politics
Tag


Top Topics Distribution (Topic Labels)
Count
120000
100000
80000
60000
40000
20000
0
119504
91251
48451
37696
28594
28554
18611
15341
14087
11889
Unknown
College
Work
Teenage
Gender
Arg
Policy
International
Sell
Travel
Topic

- Pie chart and score vs controversiality boxplot Top 8 subreddit pie chart:



The manual ought to provide the explanation that these plots assist in determining what tags to retain (educational vs non educational) and display the variation in scores with controversy.

7 Data Reduction and Creation of 'df_updated'

The current data frame 'df' had to be transformed into 'df_updated' using relevance filtering in order to be reproducible.

Columns were eliminated as redundant or largely meaningless structurally:

i. unnamed: 0
ii. comment_id
iii. comment_parent_id
iv. comment_body

These were removed by using:

4.5 Data Reduction and Relevance Filtering

```
# Drop columns with all zeros or all NaNs (you can specify which columns to drop directly too)
cols_to_drop = ['unnamed: 0', 'comment_id', 'comment_parent_id', 'comment_body']
df = df.drop(columns=cols_to_drop, errors='ignore')
```

Below image describes about the relevant tags list (which can be tailored to the future researcher) and note that tags are normalised to lower case and insensitively matched on case:

```
# Define KEEP list (relevant tags only); update this as per your research focus
relevant_tags = [
    'College', 'Work', 'International', 'Social', 'Conversation', 'Stats', 'Treatment',
    'Study', 'Education', 'School'  # ← Example: add/remove as per analysis, exclude things like 'Religion', 'Policy', etc.
]

# Make tag column lowercase and strip whitespace if needed
df['tag'] = df['tag'].astype(str).str.strip()

# Only keep rows where tag is in relevant_tags (case-insensitive matching)
df_updated = df[df['tag'].str.lower().isin([tag.lower() for tag in relevant_tags])].copy()

# Reset index for a clean dataframe after filtering
df_updated = df_updated.reset_index(drop=True)

print("Updated Tag values:", df_updated['tag'].unique())
print("Shape of df_updated:", df_updated.shape)
df_updated.head()
```

```
Updated Tag values: ['College' 'Work' 'International' 'Social' 'Stats' 'Treatment'
 'Conversation']
Shape of df_updated: (177485, 6)
```

	subreddit	body	controversiality	score	topic	tag
0	NaN	i've been a teacher for 10 years, and i don't ...	0	4402	17.0	College
1	NaN	i snatched a note from a student only to reali...	0	4339	17.0	College
2	NaN	i teach at the university level. only one thi...	0	4264	8.0	Work
3	NaN	rwandan living in us. i was 13 and in rwanda w...	0	4215	6.0	International
4	NaN	i just wanna bring to the attention here that ...	0	4190	8.0	Work

The resulting ‘df_updated’ has the shape (177485, 6) and is the base dataset of all the further analyses. Users who want to change inclusion criteria can change the relevant tags list and rerun since this step.

8 Sentiment Analysis Design

8.1 Model used and Pipeline

Hugging Face transformers.pipeline with the RoBERTa model ‘cardiffnlp/twitter-roberta-base-sentiment-latest’ is used to perform sentiment analysis.
Design:

5. Sentiment Analysis

```
import pandas as pd
from transformers import pipeline
from tqdm import tqdm
import nltk
nltk.download('punkt')

# Ensure your dataframe is ALL records
df_proc = df_updated.copy()
print("Shape of df_proc BEFORE analysis should be:", df_proc.shape)
print("Should match the shape of df_updated:", df_updated.shape)
texts = df_proc['body'].fillna('').tolist()
print("Number of texts being analyzed:", len(texts))  # This MUST equal 177485!
```

```
Shape of df_proc BEFORE analysis should be: (177485, 6)
Should match the shape of df_updated: (177485, 6)
Number of texts being analyzed: 177485
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Gaurav\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

This section applies batched sentiment analysis using a transformer pipeline to classify the sentiment of each post.

```
# Define HuggingFace pipeline and batching code
sentiment_pipe = pipeline('sentiment-analysis', model="cardiffnlp/twitter-roberta-base-sentiment-latest", device=-1)
```

8.2 Batched inference

A helper function is required to operate in batches of 16 to process data: in order to work with 177,485 texts.

```python
def batched_sentiment(texts, batch_size=16):
    labels, scores = [], []
    for i in tqdm(range(0, len(texts), batch_size), desc=f"Batched Sentiment ({len(texts)) records)"):
        batch = [str(t)[:512] for t in texts[i:i + batch_size]]
        results = sentiment_pipe(batch)
        labels.extend([r['label'] for r in results])
        scores.extend([r['score'] for r in results])
    return labels, scores
```

Python

```
WARNING:tensorflow:From C:\Users\Gaurav\AppData\Roaming\Python\Python313\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please

Some weights of the model checkpoint at cardiffnlp/twitter-roberta-base-sentiment-latest were not used when initializing RobertaForSequenceClassification: ['roberta.pooler.dense.bias',
- This IS expected if you are initializing RobertaForSequenceClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a Bert
- This IS NOT expected if you are initializing RobertaForSequenceClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceCl
Device set to use cpu
```

Output is saved in the form of new columns:

```python
# Make sure previous cells (function, pipeline) were run FIRST in your kernel!
texts = df_proc['body'].fillna('').tolist()
print("ACTUAL number of bodies to send to sentiment:", len(texts))  # Should show 177485

sent_labels, sent_scores = batched_sentiment(texts)
df_proc['sentiment'] = sent_labels
df_proc['sentiment_score'] = sent_scores
```

Python

```
ACTUAL number of bodies to send to sentiment: 177485
Batched Sentiment (177485 records): 100%|██████████| 11093/11093 [4:48:14<00:00,  1.56s/it]
```

- Total dataset 'df_updated' has 177,485 records.
- batch_size is 16.
- The tqdm progress bar is counting iterations, each iteration (batch) containing 16 records.
- On the left: '11093' denotes number of batches being proccessed so far.
- On the right: '11093' denotes number of batches required. ([177685/16] = 11093 batches)

The manual must point out the fact that truncation to 512 characters will guarantee the model with the maximum length of the sequence.

9 Ethical Position Zero-Shot Classification

9.1 Model and labels

'Facebook/bart-large-mnli' is utilized by the ethical stance pipeline as a zero-shot classifier.

6. Ethical Stance Analysis

```python
from transformers import pipeline
from tqdm import tqdm

zero_shot_pipe = pipeline('zero-shot-classification', model="facebook/bart-large-mnli", device=-1)
ethics_labels = ["prosocial", "neutral", "harmful", "ethical", "unethical", "moral dilemma"]
```

9.2 Batched zero-shot function

Since zero-shot is computationally intensive, the batch size of 8 is taken:

```python
def batched_zero_shot(texts, candidate_labels, batch_size=8):
    all_labels = []
    for i in tqdm(range(0, len(texts), batch_size), desc=f"Ethical Stance ({len(texts)} records)"):
        batch = [str(t)[:512] for t in texts[i:i+batch_size]]
        results = zero_shot_pipe(batch, candidate_labels)
        # Handle dict/list outputs from pipeline
        if isinstance(results, dict):  # very rare, just for single text as string
            results = [results]
        for res in results:
            label = res['labels'][0] if isinstance(res, dict) else res[0]['labels'][0]
            all_labels.append(label)
    return all_labels
```

Python

```python
texts = df_proc['body'].fillna('').tolist()
print("Ethical stance - records to process:", len(texts))  # Should be 177485

df_proc['ethical_stance'] = batched_zero_shot(texts, ethics_labels)
```

Python

```
Ethical stance - records to process: 177485
Ethical Stance (177485 records): 100%|██████████| 22186/22186 [91:31:54<00:00, 14.85s/it]
```

- Total dataset 'df_updated' has 177,485 records.
- batch_size here is 8.
- The tqdm progress bar is counting iterations, each iteration (batch) containing 8 records.
- On the left: '22186' denotes number of batches being proccessed so far.
- On the right: '22186' denotes number of batches required. ([177685/8] = 22186 batches)

9.3 Ensuring Checkpoint

The notebook is used to save that in-between results in order to avoid re-running long jobs:

```
df_proc.to_csv('reddit_with_sentiment_ethics.csv', index=False)
```

Python

Once I performed 'Ethical Stance Analysis' I made sure to backup my results in a csv file as observed above. This file contains

- All base dataframe columns,
- The sentiment columns (sentiment, sentiment_score),
- The ethical stance column (ethical_stance).

This file 'reddit_with_sentiment_ethics.csv' should be re-used by future users when they have access to it and loaded using pd.read_csv after which creativity metrics are done.

9.4 Visualizations for Sentiment Analysis and Ethical Stance Analysis

In this section, we have used the previously saved results for sentiment and ethical stance i.e. 'reddit_with_sentiment_ethics.csv' using seaborn library to visualize:

- Sentiment Distribution (Pie and Bar Chart)
- Ethical Stance Distribution (Pie and Bar Chart)
- Sentiment by Tag
- Ethical Stance by Sentiment

10 Creativity Quantification

10.1 Semantic diversity with Sentence-BERT 10.1

SentenceTransformer: a Sentence- Bert model (all-MiniLM-L6-v2) is loaded:

7.1 Semantic Diversity

```
from sentence_transformers import SentenceTransformer
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

# Load model (downloads on first use)
sbert = SentenceTransformer('all-MiniLM-L6-v2')

# Compute embeddings
embeddings = sbert.encode(df_proc['body'].fillna('').tolist(), convert_to_numpy=True, show_progress_bar=True)
```

A subsample of 1,000 posts is sampled, and diversity scores are calculated as 1− mean cosine similarity:

```
# Sample a fixed reference set (e.g., 1000 random posts)
np.random.seed(42)
subset_idx = np.random.choice(len(embeddings), size=min(1000, len(embeddings)), replace=False)
ref_emb = embeddings[subset_idx, :]

# Calculate diversity as (1 - average similarity to reference set)
all_sim = cosine_similarity(embeddings, ref_emb)
diversity_scores = 1 - all_sim.mean(axis=1)
df_proc['creativity_semdiv'] = diversity_scores
```

Python

Using the seaborn library, plot of a histogram is then visualized titled as 'Semantic Diversity Distribution'.

10.2 Lexical Diversity

The NLTK tokenisation is used in lexical diversity:

7.2 Lexical Diversity

```
import nltk
nltk.download('punkt', download_dir=r'C:/ltk_data')
nltk.download('punkt_tab', download_dir=r'C:/ltk_data')
```

Python

```
[nltk_data] Downloading package punkt to C:/ltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to C:/ltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
True
```

```
import nltk
nltk.data.path.clear()
nltk.data.path.append(r'C:/ltk_data')
from nltk.tokenize import word_tokenize
print(word_tokenize("This is a test sentence."))
```

Python

```
['This', 'is', 'a', 'test', 'sentence', '.']
```

```
df_proc['lexical_diversity'] = df_proc['body'].fillna('').apply(
    lambda x: len(set(word_tokenize(x))) / len(word_tokenize(x)) if x and len(x.split()) > 4 else 0
)
```

Python

The formula may be modified in the future (e.g. other thresholds or token filters).

10.3 Zero-shot creativity labels

With a new set of labels: the same facebook/bart-large-mnli model is re-used –

7.3 Zero-Shot Creativity Label

```
from transformers import pipeline
from tqdm import tqdm

zero_shot = pipeline('zero-shot-classification', model="facebook/bart-large-mnli")
creativity_labels = ["highly creative", "original", "routine", "informative", "humorous"]

def batched_zero_shot_creativity(texts, batch_size=8):
    labels = []
    for i in tqdm(range(0, len(texts), batch_size)):
        batch = [str(t)[:512] for t in texts[i:i+batch_size]]
        results = zero_shot(batch, creativity_labels)
        # Handle list/dict output
        if isinstance(results, dict):
            results = [results]
        for res in results:
            label = res['labels'][0]
            labels.append(label)
    return labels

texts = df_proc['body'].fillna('').tolist()

df_proc['creativity_label'] = batched_zero_shot_creativity(texts)
```

[18] Python

```
WARNING:tensorflow:From C:\Users\Gaurav\AppData\Roaming\Python\Python313\site-packages\tf_keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Pleas

Device set to use cpu
100%|██████████| 22186/22186 [74:40:37<00:00, 12.12s/it]
```

A batching feature analogous to the ethics feature gives each post one best label and stores it under the name creativity label. Findings are stored in the form of Reddit with “zero-shot creativity.csv”.

11 Network Analysis Setup

The filtered ‘df_updated’ data frame is used in network analysis.

11.1 Edge construction

Edges represent the co-occurrence of relationships:

```
# Build co-occurrence pairs (e.g. between 'tag' and 'topic')
edges = []
for _, row in df_updated.iterrows():
    tag = str(row['tag'])
    topic = str(row['topic'])
    subreddit = str(row['subreddit'])
    if tag != 'nan' and topic != 'nan':   # Basic check for valid values
        edges.append((f"tag:{tag}", f"topic:{topic}"))
    if tag != 'nan' and subreddit != 'nan':
        edges.append((f"tag:{tag}", f"subreddit:{subreddit}"))
    if topic != 'nan' and subreddit != 'nan':
        edges.append((f"topic:{topic}", f"subreddit:{subreddit}"))
```

11.2 Graph Design and Metrics

```
# Create the network
G = nx.Graph()
G.add_edges_from(edges)
print("Total nodes:", G.number_of_nodes())
print("Total edges:", G.number_of_edges())
```

[17] Python

```
Total nodes: 15
Total edges: 8
```

The best nodes and community sizes are printed to be checked. The centrality nodes are identified as a subgraph which is plotted with nx.draw to produce the visual that is presented in the report.

Future scholars can alter edge logic (e.g., frequency-weighted or adding other attributes) with the same configuration structure.

12 Reproducibility Extension Points and Tips

- Always fix random seeds (as in semantic diversity) in order to get the same diversity scores in each run.
- Keep interim CSVs (reddit_with_sentiment_ethics.csv, reddit_with_zero_shot_creativity.csv) in order not to run many long transformers inference tasks again.
- In modifying appropriate tags or in the addition of new labels, record the precise lists and re-run at the corresponding step.
- It is possible to use GPU acceleration by installing CUDA and passing device= 0 in the pipeline calls; it can significantly cut the runtime time in sentiment and zero-shot analysis.

13 References

Kaggle. 2023. 'How Students Use ChatGPT – Filtered Reddit Dataset.' [online] Link: https://www.kaggle.com/datasets/faizankhandeshmukh/how-students-use-chatgpt-filtered-reddit-dataset

Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., Platen P., Ma C., Jernite Y., Plu J., Xu C., Scao T. L., Gugger S., Drame M., Lhoest Q., and Rush A.. 2020. 'Transformers: State-of-the-Art Natural Language Processing.' *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.* Link: https://doi.org/10.18653/v1/2020.emnlp-demos.6

Barbieri F., Camacho-Collados J., Anke L. E., Neves L.. 2020. 'TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification.' *In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1644–1650, Online. Association for Computational Linguistics.* Link: https://doi.org/10.18653/v1/2020.findings-emnlp.148/

Lewis M., Liu Y., Goyal N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V., and Zettlemoyer L. 2020. 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.' *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.* Link: https://doi.org/10.18653/v1/2020.acl-main.703

Williams A., Nangia N., and Bowman S. 2018. 'A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.' *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.* Link: https://doi.org/10.18653/v1/N18-1101

Reimers N. and Gurevych I. 2019. 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.' *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.* Link: https://doi.org/10.18653/v1/D19-1410