


# NLP Project 02 All tasks Report

## Task 01 Encoder(Bert):

### Key Components

#### 1. Data Processing Pipeline

- **Text Cleaning:** Removes URLs, mentions, punctuation while preserving emotional content
- **Label Mapping:** Converts numeric labels (0-3) to emotion names
- **Stratified Splitting:** 80% train, 10% validation, 10% test split maintaining class balance
- **Tokenization:** Uses BERT tokenizer with dynamic padding optimized for text length

A screenshot of a Google Colab notebook interface. The browser address bar shows a Google Drive link. The notebook title is 'Task\_01\_Encoder.ipynb'. The code cell output displays a class balance analysis for four emotion classes: Anger (5816 samples, 26.5%), Joy (5793 samples, 26.4%), Neutral (5168 samples, 23.6%), and Sadness (5145 samples, 23.5%). Below this, it shows sample cleaned texts for each emotion, comparing the original text with the cleaned version where URLs, mentions, and punctuation have been removed.

```
Class balance analysis:  
Anger: 5816 samples (26.5%)  
Joy: 5793 samples (26.4%)  
Neutral: 5168 samples (23.6%)  
Sadness: 5145 samples (23.5%)  
  
Sample cleaned texts from each emotion:  
  
ANGER (Label: 3):  
Original: i dont even understand the intro to this book ...  
Cleaned: i dont even understand the intro to this book...  
  
JOY (Label: 0):  
Original: happy mothers day mommy and grandma haha ily...  
Cleaned: happy mothers day mommy and grandma haha ily...  
  
NEUTRAL (Label: 2):  
Original: tommcfly i saw you on tues and last niiiiight lt3 so amazzzing didnt even notice...  
Cleaned: tommcfly i saw you on tues and last niiiiight lt so amazzzing didnt even notice ...  
  
SADNESS (Label: 1):  
Original: not a very good day at the house...  
Cleaned: not a very good day at the house...
```

#### 2. Model Architecture

- **Base Model:** bert-base-uncased
- **Classification Head:** 4-class emotion classification
- **Training:** 4 epochs with 2e-5 learning rate, batch size 16
- **Optimization:** AdamW with weight decay, gradient clipping

#### 3. Performance Results

text

Overall Metrics:

- Accuracy: 67.03%

- F1-Score (Macro): 67.33%
- Precision: 68.21%
- Recall: 67.00%



#### ENHANCED TRAINING CONFIGURATION

##### Enhanced Training Arguments:

Learning Rate: 2e-05  
Batch Sizes: Train=16, Eval=32  
Epochs: 4  
Warmup Steps: 500  
Weight Decay: 0.01  
FP16 Enabled: True  
Max Grad Norm: 1.0  
Device: cuda

Enhanced training configuration ready with error handling!

#### Per-Class Performance:

- Anger: 60.56% F1 (Good recall: 65.00%)
- Joy: 57.88% F1 (Balanced performance)
- Neutral: 74.13% F1 (Best performing class)
- Sadness: 76.75% F1 (Excellent precision: 85.29%)

## 4. Technical Implementation

- **Framework:** Hugging Face Transformers + PyTorch
- **Tokenization:** Dynamic padding with attention masks
- **Evaluation:** Comprehensive metrics including per-class analysis
- **Visualization:** Confusion matrix and prediction examples

#### Key Strengths

1. **Robust Preprocessing:** Handles social media text with proper cleaning
2. **Class Balance:** Maintains equal distribution across all splits
3. **Comprehensive Metrics:** Beyond accuracy, includes F1, precision, recall per class
4. **Error Handling:** Training pipeline with recovery mechanisms
5. **Visual Analysis:** Confusion matrix for model behavior insight

#### Model Insights

- **Best at:** Detecting Neutral and Sadness emotions

- **Challenge:** Distinguishing between Anger and Joy (lower F1 scores)
- **Strength:** High precision for Sadness (85%) - rarely misclassifies other emotions as sadness

#### Final Test Performance:

- **Accuracy:** 67.03%
- **F1-Score (Macro):** 67.33%
- **Precision (Macro):** 68.21%
- **Recall (Macro):** 67.00%

#### Per-class Performance:

- **Anger:** F1=60.56%, Precision=56.69%, Recall=65.00%
  - **Joy:** F1=57.88%, Precision=56.85%, Recall=58.95%
  - **Neutral:** F1=74.13%, Precision=73.99%, Recall=74.27%
  - **Sadness:** F1=76.75%, Precision=85.29%, Recall=69.76%
- 

## Task 02 Decoder: (GPT2)

### Key Features

- **Data Processing:** Handles complex recipe data with titles, ingredients (NER format), and cooking instructions
- **Model Architecture:** GPT-2 (124M parameters) fine-tuned for recipe generation
- **Training Pipeline:** Complete training loop with validation and evaluation
- **Quality Assessment:** Automated recipe quality scoring and ROUGE metrics
- **Interactive Generation:** Functions for prompt-based and ingredient-based recipe creation

```
🔗 Found CSV files in Task_02_Decoder folder: ['/content/drive/MyDrive/Colab/Task_02_Decoder/dataset.csv']
```

```
Dataset loaded successfully!  
Dataset shape: (2231143, 6)  
Column names: ['title', 'NER', 'Extended_NER', 'genre', 'label', 'directions']
```

```
⇒ TOKENIZING DATASETS
Tokenizing training set...
Tokenizing train: 100% ██████████ 4000/4000 [00:06<00:00, 562.18 examples/s]
Tokenizing validation set...
Tokenizing validation: 100% ██████████ 500/500 [00:00<00:00, 615.24 examples/s]

Tokenization complete!
DatasetDict({
  train: Dataset({
    features: ['input_ids', 'attention_mask'],
    num_rows: 4000
  })
  validation: Dataset({
    features: ['input_ids', 'attention_mask'],
    num_rows: 500
  })
})
```

```
⇒ LOADING GPT-2 MODEL
Model: gpt2
Total parameters: 124,439,808
Trainable parameters: 124,439,808
Model device: cuda:0
```

```
⇒ CONFIGURING TRAINING
Training Arguments:
  Learning Rate: 5e-05
  Train Batch Size: 4
  Gradient Accumulation: 4
  Effective Batch Size: 16
  Epochs: 3
  FP16: True
```

🔄 This will take approximately 30-40 minutes on a T4 GPU...

[750/750 14:51, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	2.405700	2.238946
2	2.261000	2.172313
3	2.205100	2.155329

There were missing keys in the checkpoint model loaded: ['lm\_head.weight'].

```
=====
TRAINING COMPLETED!
=====
Training Loss: 2.5944
Training Runtime: 892.38 seconds (14.87 minutes)
Training Samples/Second: 13.45
Training Steps/Second: 0.84
```

🔄 EVALUATING ON VALIDATION SET

[125/125 00:10]

Validation Results:  
eval\_loss: 2.1553  
eval\_runtime: 10.4329  
eval\_samples\_per\_second: 47.9250  
eval\_steps\_per\_second: 11.9810  
epoch: 3.0000

```
=====
Test 1/5
=====
Prompt: Recipe: Chocolate Chip Cookies | Ingredients:
-----
Recipe: Chocolate Chip Cookies | Ingredients: ["baking soda", "water", "butter", "unsalted butter", "eggs", "milk", "salt", "unsalted butter", "vanilla", "milk", "bro
=====
Test 2/5
=====
Prompt: Recipe: Chicken Tikka Masala | Ingredients:
-----
Recipe: Chicken Tikka Masala | Ingredients: ["olive oil", "turmeric", "salt", "mushrooms", "pepper", "chicken stock", "fresh ginger", "paprika", "garlic", "hot pepper
=====
Test 3/5
=====
Prompt: Recipe: Vegetarian Pasta | Ingredients:
-----
Recipe: Vegetarian Pasta | Ingredients: ["pasta", "sugar", "chili powder", "balsamic vinegar", "salt", "egg yolks", "salt", "ground black pepper", "oil", "green onion
=====
Test 3/5
=====
Prompt: Recipe: Vegetarian Pasta | Ingredients:
-----
Recipe: Vegetarian Pasta | Ingredients: ["pasta", "sugar", "chili powder", "balsamic vinegar", "salt", "egg yolks", "salt", "ground black pepper", "oil", "green onion
=====
Test 4/5
=====
Prompt: Recipe: Banana Smoothie | Ingredients:
-----
Recipe: Banana Smoothie | Ingredients: ["brown sugar", "flour", "sugar", "flour", "egg yolks", "salt", "vanilla", "vanilla", "salt", "vanilla", "vanilla", "soda", "va
=====
Test 5/5
=====
Prompt: Recipe: Grilled Salmon | Ingredients:
-----
```

```
GENERATING FROM INGREDIENTS

Generating recipes from ingredients:
Ingredients: chicken breast, tomatoes, onions, garlic, olive oil
Recipe Name: Italian Chicken
Recipe: Italian Chicken | Ingredients: chicken breast, tomatoes, onions, garlic, olive oil | Instructions: ["Cut chicken into quarters and bring to a boil. Drain and

Ingredients: eggs, milk, flour, sugar, butter
Recipe Name: Pancakes
Recipe: Pancakes | Ingredients: eggs, milk, flour, sugar, butter | Instructions: ["Preheat oven to 350\u00b0F. Line a large baking sheet with parchment paper. In a la

Ingredients: pasta, cheese, cream, black pepper
Recipe Name: Cacio e Pepe
Recipe: Cacio e Pepe | Ingredients: pasta, cheese, cream, black pepper | Instructions: ["Preheat oven to 400\u00b0F.", "In a large bowl, whisk together pasta, cream c
```

## Performance Metrics

- **Training Loss:** 2.59 → **Validation Loss:** 2.16 (after 3 epochs)
- **ROUGE Scores:**
  - ROUGE-1: 0.242
  - ROUGE-2: 0.092
  - ROUGE-L: 0.174
- **Quality Score:** 0.87/1.00 (based on recipe structure and content)

## Model & Training

- **Base Model:** GPT-2 (124M parameters)
- **Vocabulary:** 50,257 tokens
- **Training Data:** 4,000 recipes (5,000 total dataset)
- **Hardware:** Tesla T4 GPU (15.8GB VRAM)
- **Framework:** PyTorch + Hugging Face Transformers

---

## Task 03 Encoder&Decoder(T5):

### Project Overview

This code implements a **T5-based text summarization system** using an encoder-decoder transformer architecture. The project fine-tunes the T5-small model to generate concise summaries from news articles.

### Key Components

#### 1. Data Preprocessing

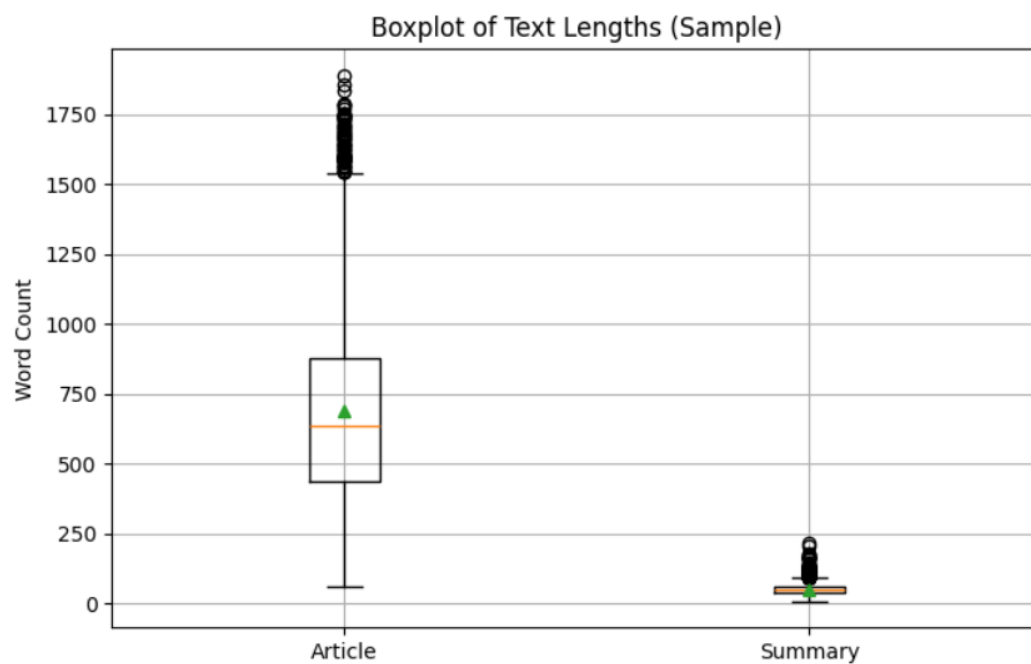
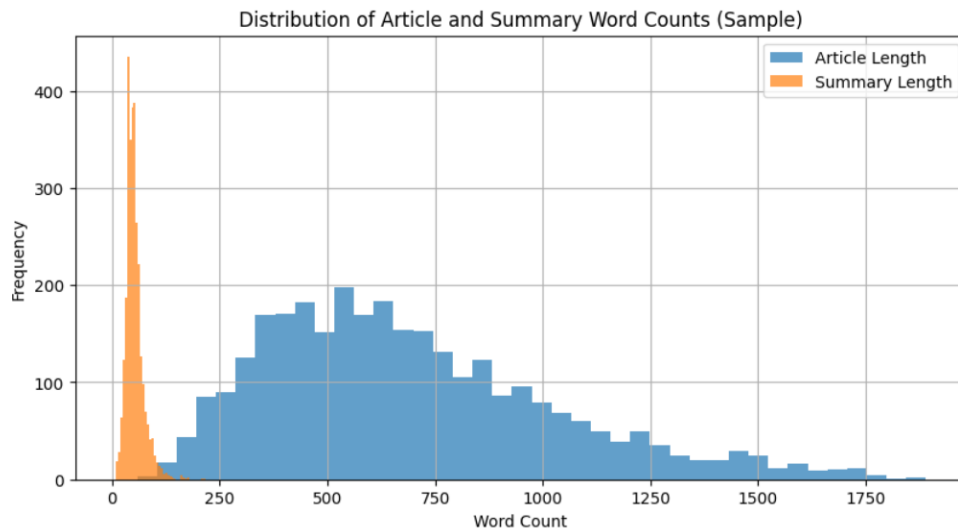
- **Data Loading:** Loads pre-split CSV files (train, validation, test) from Google Drive
- **Data Cleaning:** Removes null values, handles duplicates, standardizes column names
- **Text Analysis:** Analyzes article/summary length distributions and compression ratios
- **Sampling:** Uses 2,500 training samples for computational efficiency
- **Tokenization:** Applies T5 tokenizer with "summarize: " prefix for instruction tuning

## Images from Notebook

```
Found CSV files: ['/content/drive/MyDrive/Colab Notebooks/Project_02/Task_03_Encoder&Decoder/dataset/test.csv', '/content/drive/MyDrive/Colab Notebooks/Project_02/Task_03_Encoder&Decoder/dataset/train.csv', '/content/drive/MyDrive/Colab Notebooks/Project_02/Task_03_Encoder&Decoder/dataset/validation.csv']  
Datasets loaded successfully!  
Train shape: (287113, 3)  
Validation shape: (13368, 3)  
Test shape: (11490, 3)  
Columns: ['id', 'article', 'highlights']
```

### Compression Ratio Statistics:

```
count      3000.000000  
mean        0.091038  
std         0.053616  
min         0.007387  
25%         0.053997  
50%         0.077714  
75%         0.114512  
max         0.758621  
Name: compression_ratio, dtype: float64
```





#### TOKENIZING DATASETS (memory safe mode)

Tokenizing train: 100%  2500/2500 [00:23<00:00, 86.50 examples/s]  
Tokenizing validation: 100%  200/200 [00:02<00:00, 80.46 examples/s]  
Tokenizing test: 100%  100/100 [00:00<00:00, 135.43 examples/s]

Tokenization complete (minimal memory cache).

```
DatasetDict({
  train: Dataset({
    features: ['input_ids', 'attention_mask', 'labels'],
    num_rows: 2500
  })
  validation: Dataset({
    features: ['input_ids', 'attention_mask', 'labels'],
    num_rows: 200
  })
  test: Dataset({
    features: ['input_ids', 'attention_mask', 'labels'],
    num_rows: 100
  })
})
```

 [1500/1500 4:02:44, Epoch 1/1]

Step	Training Loss
200	4.098700
400	4.712800
600	4.746000
800	4.751400
1000	4.570900
1200	4.469800
1400	4.085700

TRAINING COMPLETED SUCCESSFULLY!

Train metrics: {'train runtime': 14574.8652, 'train samples per second': 0.103, 'train steps per second': 0.103, 'total flos': 50753175552000.0, 'train loss': 4.477307942708333, 'epoch': 1.

#### TESTING SUMMARIZATION

Sample 1/10

ARTICLE (first 300 chars):

A Hollywood-inspired experiment to help dementia patients by waking them up with video recordings from loved ones is taking place in New York. The hope is that the videos shown to residents

ORIGINAL SUMMARY:

Residents at Hebrew Care Home in New York are woken by video recordings from loved ones to help ease their confusion and agitation .  
Idea was inspired by the 2004 film 50 First Dates starring Adam Sandler .  
Experiment will be evaluated next month but results are 'very positive'

GENERATED SUMMARY:

the idea is borrowed from the 2004 Adam Sandler film 50 First Dates. a suitor, played by Sandler, uses videos to remind her of him.

Sample 2/10

ARTICLE (first 300 chars):

Eerie footage has emerged showing a mysterious 100 metre-wide black ring of smoke floating over clear skies in Kazakhstan - convincing locals it was caused by a UFO. The clip was filmed on

ORIGINAL SUMMARY:

Clip shows mysterious black cloud hanging over the village of Shortandy .  
The perfect hoop shape sat in the air not moving for more than 15 minutes .  
Eerie video has been viewed tens of thousands of times on Youtube .  
While some viewers are suggesting the cloud was an alien spacecraft, experts think it could have been caused by nearby factories .

GENERATED SUMMARY:

the clip was filmed in shortandy village, 40 miles north of Astana. it captures the cloud hovering in the sky for 15 minutes before suddenly vanishing without trace. the video has been vi

Sample 3/10

ORIGINAL SUMMARY:

Doug Hughes appeared in U.S. District Court in Washington on Thursday, one day after he steered his tiny aircraft onto the Capitol's West Lawn .  
He was charged with operating an unregistered aircraft and violating national airspace before being released on his own recognizance .  
He was sent back to his Tampa home, where he must check in weekly with authorities starting next week .

GENERATED SUMMARY:

the postal carrier who flew a gyrocopter onto the lawn of the U.S. Capitol is facing two criminal charges. but he's being released from federal custody to return to Florida.

Sample 4/10

ARTICLE (first 300 chars):

Samantha Crossland escaped a custodial sentence after the £22,000 she admitted stealing from her employer and friend had been repaid in full . The daughter of a millionaire lottery winning

ORIGINAL SUMMARY:

Samantha Crossland admitted stealing more than £22,000 from her friend .  
The 30-year-old was a manager at the Child's Play nursery in Dewsbury .  
She took the money from a cash box containing the children's fees .  
Crossland avoided a custodial sentence after the money was repaid in full .

GENERATED SUMMARY:

Samantha Crossland, 30, was pocketing parents' fees being paid to nursery. she was told she faced custodial sentence unless money was paid back. she admitted stealing from her employer and

Sample 5/10

ARTICLE (first 300 chars):

A former Guantanamo Bay prisoner arrested over the terrorist killing of a top female prosecutor in Uganda had been awarded £1million in compensation by the British government following his

## 2. Model Configuration

- **Architecture:** T5-small (60M parameters) encoder-decoder transformer
- **Memory Optimization:**
  - Gradient checkpointing enabled
  - Attempts float16 precision
  - Reduced sequence lengths (128 input, 32 output tokens)
- **Device Management:** Automatic CPU/GPU detection

## 3. Training Setup

- **Data Collator:** Dynamic padding for efficient batching
- **Evaluation Metrics:** ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum)
- **Memory Safety:** Small batch sizes and careful dataset handling

## 4. Key Features

- **Task Prefix:** Uses "summarize: " prefix for T5 instruction following
- **Progressive Sampling:** Reduces dataset sizes to manage memory constraints
- **Comprehensive Evaluation:** Implements full ROUGE metric computation
- **Error Handling:** Fallback mechanisms for memory-intensive operations

## Workflow Summary

1. **Data Preparation** → Load and clean news article datasets
2. **Text Analysis** → Understand length distributions and compression patterns
3. **Tokenization** → Convert text to T5-compatible token IDs
4. **Model Setup** → Configure T5 with memory optimizations
5. **Training Ready** → Prepare datasets, collator, and evaluation metrics
6. **Evaluation Framework** → Implement ROUGE scoring for summary quality

## Technical Highlights

- **Memory Management:** Multiple strategies to handle large text datasets

- **Transformer Fine-tuning:** Adapts pre-trained T5 for summarization task
  - **Quality Metrics:** Uses standard NLP evaluation metrics (ROUGE)
  - **Scalable Design:** Can be extended to larger models and datasets
-