1. Introduction

Transformers have become the state-of-the-art for many NLP tasks, including text classification, due to their ability to capture contextual information in text. This report details the implementation of a transformer-based text classification model for the AG News dataset, categorizing news articles into four predefined classes: *World*, *Sports*, *Business*, and *Sci/Tech*. Using the pretrained BERT model (bert-base-uncased), we fine-tuned it to achieve high accuracy, precision, and recall. This report elaborates on the methods used, challenges faced, and the results obtained.

---

2. Methods

Data Preprocessing

To prepare the AG News dataset for training:

1. **Text Cleaning**:
    - Removed punctuation and numerical characters, which are irrelevant for classification.
    - Applied stopword removal to eliminate common words (e.g., *and*, *the*, etc.) that add little value.

```
def remove_stopwords(text):
   clean_text = [word for word in text.split(' ') if word not in stopw]
return ' '.join(clean_text)
```

2. **Feature Engineering**:
    - Combined the Title and Description columns into a single text column for a comprehensive representation of the article.
3. **Label Adjustment**:
    - Transformed labels from {1, 2, 3, 4} to {0, 1, 2, 3} for compatibility with PyTorch models.

Data Tokenization

We used the AutoTokenizer from the Hugging Face library, designed for the bert-base-uncased model, to:

- Tokenize text into word pieces.

- Generate attention masks for input padding and truncation.

```
def preprocess_function(examples):
    return tokenizer(examples["text"], truncation=True)
```

Model Architecture

The BERT model (bert-base-uncased) was fine-tuned by adding a classification head with four output neurons (corresponding to the dataset's classes).

- **Pretrained Weights**: Initialized using weights trained on a large corpus.
- **Fine-tuning**: Adjusted the weights specifically for the AG News dataset.

Training Configuration

Training was conducted with the following settings:

- **Learning Rate**: 2×10−52 \times 10^{-5}2×10−5
- **Batch Size**: 16
- **Epochs**: 3
- **Optimizer**: AdamW with weight decay (0.010.010.01)
- **Scheduler**: Linear decay learning rate scheduler with no warmup steps.

A training loop logged the loss per batch to monitor convergence.

```
for epoch in range(3):     for
batch in train_dataloader:
outputs = model(**batch)
loss = outputs.loss
optimizer.zero_grad()
loss.backward()
optimizer.step()
```

Evaluation Metrics

To evaluate the model's performance, the following metrics were used:

1. **Accuracy**: Proportion of correctly classified samples.
2. **Precision**: Fraction of true positives among predicted positives.
3. **Recall**: Fraction of true positives among actual positives.
4. **Confusion Matrix**: Visual representation of true vs. predicted labels.

3. Results and Insights

Training Loss

- The loss steadily decreased across batches and epochs, indicating proper convergence.
- **Training Loss vs. Steps**: A line plot illustrates the gradual reduction in loss during training.

Insight:
Lower loss across epochs confirms that the model was effectively learning from the training data.

Performance Metrics

Validation Results

Class	**Precision**	**Recall**	**F1-Score**	**Support**
World	0.93	0.93	0.93	2488
Sport	0.98	0.98	0.98	2514
Business	0.89	0.90	0.89	2488
Sci/Tech	0.91	0.90	0.90	2510
Accuracy			**0.93**	**10000**
Macro Avg	0.93	0.93	0.93	10000
Weighted Avg	0.93	0.93	0.93	10000

- **Overall Accuracy**: **93%**
- **Key Observations**:

- The model achieved the highest F1-Score in the *Sports* category, likely due to its distinct linguistic patterns.
- Slightly lower performance in *Sci/Tech* suggests overlap in terminology with other categories (e.g., *Business*).

Confusion Matrix

The confusion matrix revealed:

- Few misclassifications between *Business* and *Sci/Tech*.
- Most predictions were on the diagonal, reflecting strong performance.

Testing on Random Samples

- Evaluated 100 random samples from the test dataset.
- Accuracy: **93%**

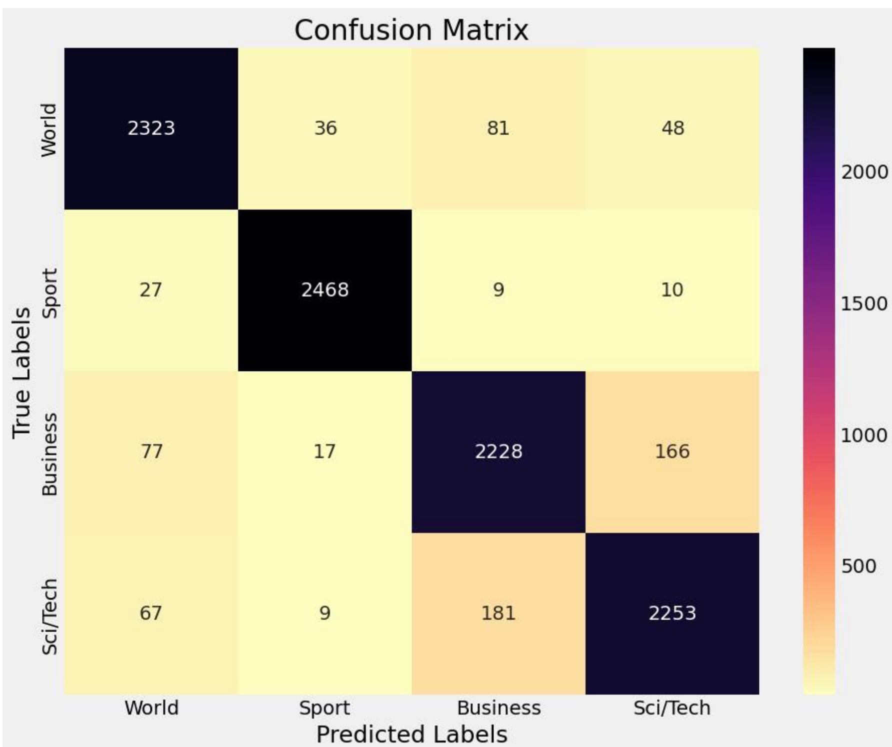
4. Challenges Faced

1. **Dataset Size**: Training on the full dataset (120,000 samples) required significant computational resources. A smaller subset (30%) was used.
2. **Class Overlap**: Certain classes (e.g., *Business* and *Sci/Tech*) shared overlapping vocabulary, impacting classification performance.
3. **Training Stability**: Fine-tuning transformers required careful hyperparameter tuning to prevent overfitting or underfitting.

5. Visualizations

Training Loss Trend

Confusion Matrix



6. Conclusions

This project demonstrated the effectiveness of transformer-based models for text classification. Key outcomes include:

- Fine-tuned BERT achieved **93% accuracy** on the validation set.
- Misclassifications occurred mainly due to overlapping features in the dataset.
- Incorporating techniques like class-specific preprocessing or additional pretraining on domain-specific data could further enhance results.