**Comparative Evaluation of Transformer-Based Models for Text Classification**

**Francky Ronsard Saah, 211307100**  
Department of Information Systems Engineering  
Kocaeli University, Kocaeli, Turkey  
[francky877832@gmail.com](mailto:francky877832@gmail.com)

**Abstract**

This project focuses on developing a machine learning solution to categorize technical articles into five specialized domains: Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, and Large Language Models (LLM). A total of 30,506 articles were collected from arXiv.org and processed using natural language techniques such as tokenization, lemmatization, and stopword removal. Class balancing, duplicate removal, and visualization techniques were applied to ensure the dataset's quality. Five transformer-based models—BERT, RoBERTa, DeBERTa, GPT, and ALBERT—were fine-tuned and evaluated. Among them, DeBERTa achieved the best overall performance, while ALBERT stood out for fast inference. This report presents the complete workflow, evaluation results, and a comparative analysis.

**Keywords**—Article Classification, Transformers, Text Preprocessing, Model Evaluation, NLP, Machine Learning

**1. Introduction**

In the digital age, managing and organizing massive volumes of technical literature has become increasingly vital. This project addresses the task of automatic classification of scholarly articles into five key domains using transformer-based NLP models. The process includes data acquisition, preprocessing, visualization, and classification using state-of-the-art transformers. The report details the challenges encountered and methods applied, particularly focusing on optimizing classification performance.

**2. Data Collection and Preprocessing**

**2.1 Data Sources**

Articles were sourced from [arXiv.org](https://arxiv.org/), manually curated for relevance. The raw dataset contained 30,506 entries, distributed across five categories:

* Deep Learning: 6,485
* Wireless Communication: 6,163
* Cloud Computing: 5,740
* Virtual Reality: 6,404
* Large Language Models (LLM): 5,714

Each category was downsampled to 5,000 articles, producing a balanced dataset of 25,000 records.

**2.2 Tools and Environment**

* **Language:** Python 3.9
* **Environment:** Google Colab
* **Libraries:** nltk, pandas, seaborn, matplotlib, requests, os, time

**2.3 Dataset Features**

Each article record includes:

* Title
* Summary
* URL
* Category Label

**2.4 Challenges and Solutions**

* **Class Imbalance:** Resolved via downsampling to equalize class representation.
* **Duplicate Removal:** Applied textual similarity and uniqueness filtering.
* **API Rate Limits:** Managed arXiv limitations with looping and batching queries.
* **Low Sample Volume for LLM:** Addressed by expanding and diversifying search keywords.

**2.5 Preprocessing Techniques**

Performed using NLTK, including:

* Tokenization
* Stopword Removal
* Lemmatization
* Lowercasing
* Removal of non-alphabetic characters
* Final duplicate filtering

Final result: 25,000 clean, balanced records.

**2.6 Data Visualization**

**Vefor and after with description**

* **Class Distribution Bar Plot**
* **Class Proportion Pie Chart**
* **Text Feature Correlation Heatmap**
* **Word Clouds** (highlighting terms like "network", "data", "model", etc.)

**2.7 Dataset Access**

All files have been uploaded to the instructor's Google Drive folder:  
🔗 [Dataset & Code](https://drive.google.com/drive/u/0/folders/1OqGiTY1ovuZrdBtFKZ9oRBIulh2Bmyf4)

**3. Transformer Models**

**3.1 BERT**

**Description**

* Pretrained: BERT-base-uncased
* Fine-tuned on balanced data (batch size 16, 4 epochs)
* Consistent loss drop and high stability

**3.2 RoBERTa**

* Pretrained: RoBERTa-base
* Outperformed BERT slightly in F1-score
* Slightly higher training time

**3.3 DeBERTa**

* Pretrained: DeBERTa-base
* Best overall results across all metrics
* Strong performance on imbalanced features

**3.4 GPT (DistilGPT2)**

* Fine-tuned for classification
* Lower performance but useful as baseline

**3.5 ALBERT**

* Pretrained: ALBERT-base-v2
* Fastest inference time
* Slightly lower accuracy, ideal for lightweight applications

**4. Evaluation Metrics**

* **Accuracy**
* **Precision**
* **Recall**
* **F1 Score**
* **AUC (ROC Area)**
* **Training Time (minutes)**
* **Inference Time (ms/sample)**

**5. Experimental Results**

**Table 1: Model Performance Comparison**

| **Model** | **Accuracy** | **F1 Score** | **Precision** | **Recall** | **AUC** | **Train Time (min)** | **Inference Time (ms)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BERT | 87.2% | 86.9% | 87.5% | 86.3% | 0.91 | 22 | 35 |
| RoBERTa | 88.5% | 88.2% | 89.0% | 87.5% | 0.92 | 25 | 38 |
| DeBERTa | **89.7%** | **89.5%** | 90.1% | 89.0% | 0.93 | 28 | 40 |
| GPT | 84.0% | 83.5% | 85.0% | 82.0% | 0.88 | 30 | 80 |
| ALBERT | 86.0% | 85.7% | 86.5% | 85.0% | 0.90 | 18 | **22** |

**Interpretation**

**Figures Included:**

* ROC Curves for all models + interoretation
* Confusion Matrices
* Epoch vs. Loss (Train/Validation)

**6. Discussion**

DeBERTa demonstrated the highest classification performance, validating its advanced attention mechanisms. While GPT lagged in performance, ALBERT’s low inference time makes it a practical model for real-time use cases. Each model showed different strengths based on speed, memory, and generalization.

**7. Conclusion**

This report demonstrates the complete pipeline from data acquisition to model evaluation for technical article classification. DeBERTa is the most performant model for this task, whereas ALBERT is best for lightweight applications. Future improvements include domain-specific fine-tuning and deployment optimization.

**References**

[1] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805, 2018.  
[2] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", arXiv:1907.11692, 2019.  
[3] P. He et al., "DeBERTa: Decoding-enhanced BERT with Disentangled Attention", arXiv:2006.03654, 2020.  
[4] R. Sanh et al., "DistilGPT2", Hugging Face, 2019.  
[5] Z. Lan et al., "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", arXiv:1909.11942, 2019.

**Appendix**

* Dataset and scripts: [Google Drive](https://drive.google.com/drive/u/0/folders/1OqGiTY1ovuZrdBtFKZ9oRBIulh2Bmyf4)
* Preprocessing scripts, evaluation plots, model checkpoints