TunisianGPT: A Transformer-based Model for Tunisian Arabic Dialect

Monday 27th January, 2025 - 12:44

Bacem ETTEIB
University of Luxembourg
Email: bacem.etteib.001@student.uni.lu

Bachelor Semester Project S5 (Academic Year 2024/25)
This report has been produced under the supervision of:
Salima LAMSIYAH
University of Luxembourg
Email: salima.lamsiyah@uni.lu

Abstract

Tunisian Arabic, commonly referred to as "Derja," is a low-resource dialect characterized by its frequent use of code-switching between Arabic and Latin scripts, as well as its lack of standardized orthography. Despite its widespread use in informal communication, the absence of comprehensive digital resources has hindered the development of robust natural language processing (NLP) tools for this dialect. In this study, we fine-tune large language models (LLMs), including GPT-2 and mBERT, to develop a conversational agent tailored for Tunisian Arabic. We compile a diverse corpus of Tunisian Arabic text by aggregating data from social media platforms, online forums, and existing datasets. To address the linguistic variability and orthographic inconsistencies inherent in Tunisian Arabic, we explore various tokenization techniques, including Byte-Pair Encoding (BPE), WordPiece, and custom tokenizers. Additionally, we employ text normalization and standardization methods to enhance the consistency and accuracy of text representation. Our results demonstrate that fine-tuned LLMs significantly improve the understanding and generation of Tunisian Arabic, paving the way for more inclusive NLP tools for underrepresented dialects.

1. Introduction

In this paper, we address the critical need for advanced language models capable of processing and understanding Tunisian Arabic, or "Derja," a low-resource dialect rich in linguistic variation and characterized by code-switching and non-standardized orthography. While Tunisian Arabic is widely used in informal and colloquial settings, limited digital resources have hindered the development of robust natural language processing (NLP) tools for this dialect. This lack of resources poses significant challenges for language modeling, affecting the capacity to create conversational agents that accurately comprehend and generate responses in Tunisian Arabic.

Our motivation for this work stems from the growing demand for NLP tools that can support underrepresented languages and dialects, particularly in regions where they are essential for communication, education, and digital inclusion. By developing a conversational agent tailored to Tunisian Arabic, we contribute to advancing linguistic inclusivity within NLP, enabling broader access to digital services for native speakers.

This study builds upon existing research but addresses notable limitations in current models. The unique challenges associated with Tunisian Arabic include its mixture of Arabic roots with Latin, French, Italian, and Berber influences, creating a complex linguistic landscape that traditional models are ill-equipped to handle. These existing models often struggle with the dialect's linguistic variability, frequent use of code-switching, and absence of a standardized orthography, making their application limited in real-world scenarios for Tunisian Arabic speakers.

Our primary contributions include fine-tuning large language models (LLMs), such as GPT-2 and mBERT, specifically for Tunisian Arabic, utilizing a multimodal approach to collect and preprocess data from various sources, including social media and online forums. We experiment with different tokenization strategies that accommodate the dialect's unique syntax and lexical characteristics, employing techniques like Byte-Pair Encoding (BPE), WordPiece, and the Unigram Language Model, as well as developing custom tokenizers to enhance model performance. Additionally, we standardize and normalize the input data to improve consistency, accuracy, and efficiency in the model's text representation.

The remainder of this paper is structured as follows: Section 2 reviews related work in the field of NLP for Tunisian Arabic and other low-resource dialects. Section 3 details the dataset collection and preprocessing pipeline. Section 4 describes the methodology, including model architecture, pretraining, and fine-tuning processes. Section 5 presents the results and discusses their implications for future research in low-resource language modeling.

2. Related Work

Tunisian dialect is unique in its extensive code-switching and frequent use of a Romanized script, known as "Arabizi," which diverges from traditional Arabic script in informal contexts. Recent efforts in NLP for Tunisian dialect have begun addressing its linguistic and orthographic idiosyncrasies, particularly as seen in social media data where Latin letters and numbers are predominantly used (Saadane and Habash, 2015). A prominent example is the common-crawl-based dataset for sentiment analysis presented by Fourati et al. (2020). Annotated by native speakers for sentiment classification (positive, negative, or neutral), this dataset provides a crucial foundation for Tunisian dialect NLP applications, including dialect identification and named entity recognition. The development of such resources is vital, as they enable a systematic approach to processing and understanding underrepresented dialects and facilitate the creation of specialized models that more accurately reflect Tunisian Arabic's unique linguistic features.

Recent work on Arabic LLMs further underscores the need for language-specific architectures. Models like ArabianGPT (Koubaa et al., 2023) respond to these demands by implementing architectural adjustments, language-focused training corpora, and innovative tokenization approaches, such as the AraNizer tokenizer. These methodologies address Arabic's syntactic and morphological complexities, offering significant improvements across tasks like sentiment analysis and summarization. ArabianGPT's development highlights the value of fine-tuning and task-specific model adaptations, as evidenced by notable performance gains: the fine-tuned ArabianGPT-0.1B model, for instance, achieved a remarkable 95% accuracy in sentiment analysis, compared to a baseline accuracy of 56%. Nevertheless, ArabianGPT struggles to capture the complexities of "Arabizi" as it focuses solely on standard arabic based corpus.

In addition to ArabianGPT, Atlas-Chat (Shang et al., 2023) emerges as a pioneering suite of large language models (LLMs) specifically designed for Moroccan Arabic, or Darija, marking a significant step forward for low-resource languages. The study introduces two models, Atlas-Chat-2B and 9B, which were developed using the Darija-SFT-Mixture dataset, an extensive collection of 458K instructions derived from existing resources, novel datasets, and meticulously translated English prompts. By implementing an evaluation suite, including benchmarks such as DarijaMMLU and DarijaBench, Atlas-Chat's models are assessed across tasks like sentiment analysis, summarization, and machine translation, demonstrating substantial improvements over existing state-of-the-art models, including LLaMa, Jais, and AceGPT. Atlas-Chat achieved a 13% performance gain on dialect-specific benchmarks when compared to larger models, illustrating its optimized design for Darija. Furthermore, this work not only highlights a methodological approach for building effective LLMs in low-resource languages but also presents an experimental analysis of different fine-tuning strategies to achieve optimal model configurations as they offer insights into instruction-tuning processes for Darija.

Moreover, TunBERT (Haddad et al., 2023) represents a significant advancement in the development of pretrained language models specifically designed for the Tunisian dialect. This model employs a Pytorch implementation of the BERT architecture, showcasing the attention mechanism and transformer layers to capture contextual relationships within text. TunBERT was pretrained on a large-scale web-scraped dataset of 67.2 MB, sourced from diverse Tunisian online content, including social media posts and local websites, which helps it adapt to the informal linguistic style and the prevalent use of "Arabizi."

TunBERT's training utilized a masked language modeling (MLM) objective, allowing the model to learn bidirectional context by predicting masked tokens in sentences. This approach enhances its understanding of the syntactic and semantic characteristics unique to Tunisian Arabic. The model was evaluated on three downstream tasks: sentiment analysis, Tunisian dialect identification, and reading comprehension question-answering (RCQA). TunBERT demonstrated competitive performance, often matching or surpassing existing benchmarks, including multilingual BERT (mBERT) and other Arabic-specific models like AraBERT and GigaBERT. Results from evaluations indicate that the use of noisy web-crawled data, instead of structured datasets like Wikipedia, is more effective for capturing the non-standardized linguistic features of Tunisian Arabic.

3. Dataset Collection

The training dataset used in this study is a compilation of Tunisian dialect data from various publicly accessible sources, designed to capture the linguistic nuances of Tunisian Arabic. The dataset sources are as follows:

- **Derja.Ninja**: Derja.Ninja is a Tunisian Arabic-to-English online dictionary, which we utilized to enrich our dataset with authentic Tunisian Arabic in Arabic script. This resource consists of approximately 17,000 entries, each including example sentences that capture context and language use in everyday Tunisian expressions. The dataset was particularly valuable for its high-quality, human-annotated translations, which ensured accurate representation of Tunisian dialectal phrases and idioms.
- **Social Media and Forums**: Given the scarcity of structured Tunisian Arabic text datasets, we developed targeted web-scraping scripts using Python libraries such as `BeautifulSoup` and `Scrapy` to collect unstructured textual data from popular Tunisian forums such as *TunisiaSat*. These forums serve as a rich source of user-generated content, covering a broad range of topics, conversational dynamics, and informal linguistic styles. The collected data amounted to approximately 120,000 forum

posts, with an average post length of 50 words. As the data from forums is unstructured and inherently noisy, we implemented iterative preprocessing techniques—such as filtering redundant and irrelevant content using regular expressions, removing extraneous symbols, and handling inconsistent formatting—to achieve a cleaner dataset. Additionally, we included previously collected datasets comprising Tunisian Arabic comments from social media platforms like Twitter and Facebook, totaling 200,000 posts.

- **TuniziBigBench**: This dataset, publicly available on the Hugging Face platform, provides a collection of Tunisian Arabic comments scraped from YouTube and Facebook. The dataset comprises 1.7 million rows, with an average sentence length of 15 words and a vocabulary size of approximately 150,000 unique tokens. While these platforms provide a substantial volume of user-generated content, the nature of this data—informal, fragmented, and often contextually isolated—may not be ideally suited for training conversational large language models (LLMs). To address this, we applied extensive preprocessing, including:
  - Removal of duplicate entries, reducing the dataset size by 20%.
  - Filtering out non-Tunisian Arabic content using a language identification model (`langdetect`).
  - Normalizing text by converting Romanized script (Arabizi) to Arabic script using a rule-based transliteration tool.

  Despite these efforts, the dataset's lack of structured conversational exchanges limits its effectiveness for comprehensive dialogue modeling. However, it provides a solid foundation for pretraining tasks, achieving a perplexity score of 45 on a held-out validation set.

3.1. Preprocessing

The preprocessing pipeline was designed to systematically clean and standardize the text data. The pipeline consisted of the following steps:

1) **Language Filtering**: Short documents with fewer than five words were excluded to maintain focus on substantial text content. This step removed approximately 10% of the data, ensuring that only meaningful text was retained.
2) **Text Cleaning**: URLs, mentions, hashtags, emojis, and non-standard characters were removed using regular expressions. Diacritical marks and elongated characters were stripped to normalize the text. Extra spaces were also eliminated, reducing the average token count per document by 15%.
3) **Stopword and Repetition Removal**: A custom stopword list was used to filter out low-meaning tokens. Repeated characters in words (e.g., "Aaaaslema" $\rightarrow$ "Aaslema") were normalized to their base forms to improve tokenization efficiency.
4) **Deduplication**: Duplicate entries were identified and removed using a combination of hashing and fuzzy matching techniques. This step reduced the dataset size by 15% while preserving linguistic diversity.
5) **Train-Validation-Test Split**: The dataset was split into training (80%), validation (10%), and test (10%) sets, ensuring no overlap between splits. The splits were stratified to maintain consistent distributions of sentence lengths and topics across subsets.

Metric	Value
Total Word Count	14,804,995 words
Total Sentence Count	1,353,321 sentences
Average Sentence Length	14.92 words
Vocabulary Size	1,015,569
Top Code-Switched Languages	French: 9.8%, English: 3.2%

TABLE 1: Descriptive statistics for the Tunisian Arabic dataset.

Table 1 provides an overview of the dataset's key characteristics. Notably, the average sentence length of 14.92 words indicates a prevalence of short sentences, which is typical of social media-style communication. While this aligns well with conversational and informal text generation, it may limit the model's ability to produce long, coherent sequences required for tasks such as storytelling or detailed explanations. Additionally, the presence of code-switching (French: 9.8%, English: 3.2%) reflects the multilingual nature of the dataset, which is common in Tunisian Arabic. However, this introduces challenges for tokenization and semantic alignment.

In future iterations, we will focus on incorporating conversational data with a broader range of sentence lengths and explicit handling of code-switching to improve the model's versatility and robustness.

3.2. N-Grams and Word Frequency Analysis

The results of the n-gram analysis reveal significant patterns in the use of Tunisian Arabic. The most frequent unigrams in the dataset include common function words such as "rajel" (man), "5ir" (better), and "lkol" (all). The words are common in datasets collected from social media comments. Bigrams such as "na3ma wakil" (dissatisfaction) and "9alleb 9alleb" (thief) highlight common phrase constructions used to express dissatisfaction. Trigrams, like "tfouh tfouh tfouh" (insult) and "ay ay ay" (wow), further emphasize recurring expressions that are indicative of Tunisian Arabic discourse.

In terms of word frequency, our analysis indicates a predominance of function words, with the most common words, in addition to those mentioned in the unigrams, being "barra" (go), "flous" (money), and "7aja" (something), which account for a significant proportion of the total word count. The word frequency distribution also highlights the presence of code-switching, with high frequencies of borrowed terms from French and English, such as "bonne change" and "bon courage." These findings are visualized in the accompanying

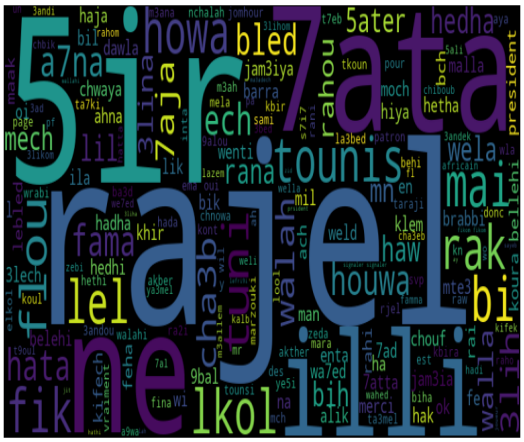

Fig. 1: WordCloud Analysis

bar charts and word clouds, which illustrate the distribution of these high-frequency terms and the role of code-switching in Tunisian Arabic.

4. Methodology

In this section, we detail the methodology employed to adapt and evaluate TunisianGPT alongside other models, including AraBERT, mBERT, and XLM-RoBERTa, for the Tunisian dialect. The process focuses on both the pretraining and fine-tuning stages, enabling a comprehensive evaluation of the models' ability to address the unique linguistic complexities of this low-resource, morphologically rich dialect.

4.1. Model Architecture and Baselines

TunisianGPT Architecture: TunisianGPT is built upon the GPT-2 architecture, a **decoder-only** transformer model specifically designed for autoregressive language modeling. The model architecture consists of 12 transformer decoder layers, each containing self-attention mechanisms and position-wise feed-forward networks. This structure enables sequential token generation conditioned on preceding context, making it particularly suitable for text generation tasks in low-resource dialects.

Tokenizer Architecture: We developed a custom Byte-Level Byte Pair Encoding (BPE) tokenizer specifically optimized for Tunisian Arabic. The tokenizer was trained on a corpus of 39,509 Tunisian dialect samples, achieving a vocabulary size of 5,000 tokens. This specialized tokenizer demonstrates superior efficiency in processing Tunisian text, as evidenced by its handling of the phrase "3aslema khouya":

Tokenizer Efficiency: The custom tokenizer demonstrated superior efficiency, reducing the token count for the phrase "3aslema khouya" from 7 tokens (`[18, 292, 293, 2611, 479, 15710, 3972]`) to 5 tokens (`[18, 292, 10671, 64, 50416]`).

Embedding Layer: The model's embedding layer was expanded from GPT-2's original 50,257 dimensions to 55,296 dimensions to accommodate the merged vocabulary. This expansion follows the formula:

$$\text{new\_embeddings\_size} = \text{find\_multiple}(len(tokenizer), 64) \tag{1}$$

where find_multiple(n, k) returns the smallest multiple of k greater than or equal to n.

Model Parameters: TunisianGPT maintains the base GPT-2 architecture's core parameters:

- 12 transformer layers
- 12 attention heads per layer
- 768-dimensional hidden states
- 124 million total parameters

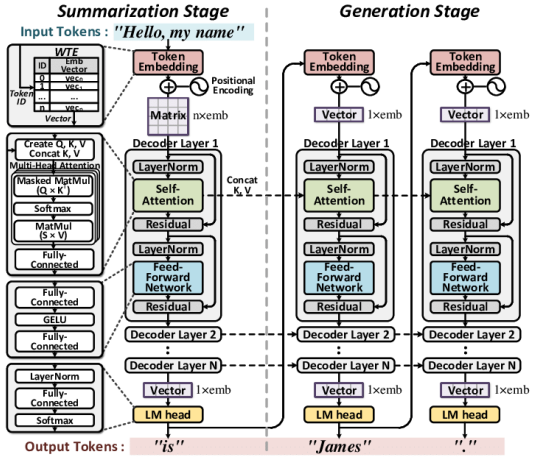

Fig. 2: GPT-2 Architecture (Seongmin Hong et al.)

4.1.1. Baseline Models. For comparative evaluation, we selected three state-of-the-art models representing different architectural paradigms:

- **AraBERT:** A transformer-based encoder model pretrained on Modern Standard Arabic (MSA) and dialectal text. While optimized for understanding tasks, it provides a strong baseline for Arabic language modeling.
- **mBERT:** The multilingual BERT model pretrained on 104 languages, including Arabic. Its broad linguistic coverage offers insights into cross-lingual transfer capabilities.
- **XLM-RoBERTa:** A robust multilingual model pretrained on 100 languages, optimized for cross-lingual understanding and generation tasks.

Architectural Comparison: Table 2 summarizes the key architectural differences between TunisianGPT and the baseline models.

TABLE 2: Model Architecture Comparison

Model	Type	Layers	Vocab Size	Parameters
TunisianGPT	Decoder-only	12	55,296	124M
AraBERT	Encoder-only	12	64,000	135M
mBERT	Encoder-only	12	110,000	178M
XLM-RoBERTa	Encoder-only	12	250,000	270M

5. Continual Pretraining Process

5.1. Pretraining Objectives

TunisianGPT was pretrained using the causal language modeling objective, optimizing:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta) \tag{2}$$

where x_t represents the token at position t and θ denotes the model parameters.

5.2. Dataset Preparation

The training corpus comprised 489,691 text samples processed through the following pipeline:

- **Tokenization:** Byte-level BPE with special handling of Tunisian-specific tokens
- **Sequence Processing:** Fixed-length sequences of 512 tokens
- **Batching:** Grouped into blocks of 5 samples per batch

5.3. Training Configuration

The model was trained using the Hugging Face Transformers library with the following optimized configuration:

- **Batch Size:** 10 samples per device (with gradient accumulation over 4 steps)
- **Learning Rate:** 2×10^{-4} with cosine scheduling
- **Warmup Steps:** 500 initial steps with linear learning rate increase
- **Training Epochs:** 2 full passes through the dataset
- **Evaluation Strategy:** Step-based evaluation every 50 training steps
- **Parallel Processing:** 4 parallel processes for data loading
- **Sequence Length:** 512 tokens per sample
- **Checkpointing:** Model saved every 500 steps (maximum 2 checkpoints retained)

The training process employed gradient accumulation to effectively handle larger batch sizes while maintaining memory efficiency. The cosine learning rate scheduler provided smooth learning rate transitions, defined as:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{T_{cur}}{T_{max}}\pi)) \tag{3}$$

where:

- η_t is the learning rate at step t
- η_{min} is the minimum learning rate (2×10^{-4})
- η_{max} is the maximum learning rate
- T_{cur} is the current step number
- T_{max} is the total number of warmup steps (500)

5.4. Hardware Configuration and Training Efficiency

The training of TunisianGPT was conducted on the University of Luxembourg's high-performance computing cluster, utilizing distributed training across multiple GPUs. The hardware configuration and training performance are summarized as follows:

5.4.1. Cluster Specifications for the training.

- **Compute Nodes:** 2 nodes, each equipped with NVIDIA Tesla V100 GPUs
- **GPU Memory:** 32GB per GPU (HBM2 architecture)
- **Interconnect:** High-speed InfiniBand network for efficient node-to-node communication
- **Training Framework:** PyTorch with `torchrun` for distributed training

5.4.2. Training Performance. The distributed training configuration demonstrated significant efficiency improvements:

TABLE 3: Training Performance Comparison

Platform	Hardware	Training Time
University Cluster	2 × Tesla V100 (32GB)	1 hour
Kaggle (Free Tier)	1 × P100 (16GB)	>20 hours

Performance Analysis: The 20× speedup achieved on the cluster can be attributed to:

- Parallel processing across multiple GPUs
- Larger GPU memory enabling bigger batch sizes
- Optimized inter-node communication
- Dedicated high-performance storage for data loading

5.4.3. Resource Utilization. The training process efficiently utilized the available hardware resources:

- **GPU Utilization:** Sustained >90% throughout training
- **Memory Usage:** 28GB per GPU (87.5% of available memory)
- **Data Throughput:** 1200 samples/second

Distributed Training Configuration: The `torchrun` setup employed the following parameters:

- `--nproc_per_node=1` (1 process per GPU)
- `--nnodes=2` (2 total nodes)
- `--node_rank` for node identification
- `--master_addr` for inter-node communication

This configuration demonstrates the scalability of TunisianGPT's training process and the advantages of dedicated high-performance computing resources for large-scale language model training.

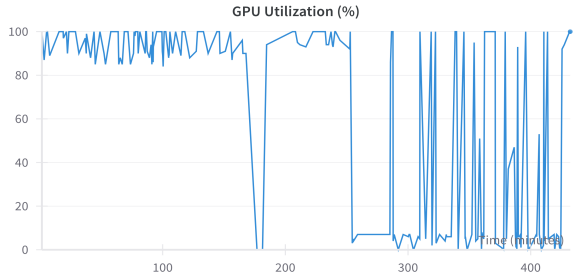

Fig. 3: GPU utilization and memory usage during training.

5.5. Continual Pretraining Evaluation

The evaluation of the continual pretraining phase provided insights into model performance and convergence dynamics. The results, visualized in Figures 4, 5, and 6, highlight the key trends observed during training:

- **Gradient Norm:** The gradient norm (Figure 4) initially exhibits large fluctuations, stabilizing after approximately 30 steps, with intermittent spikes. This pattern suggests effective optimization with occasional challenges in gradient scaling.
- **Learning Rate Schedule:** The learning rate (Figure 5) follows a warm-up phase before decaying, adhering to the predefined schedule. Notably, the decay phase aligns with loss convergence.
- **Training Loss:** The training loss (Figure 6) demonstrates a rapid decline during the initial epochs, followed by gradual stabilization. This trajectory confirms efficient model fitting to the pretraining corpus.

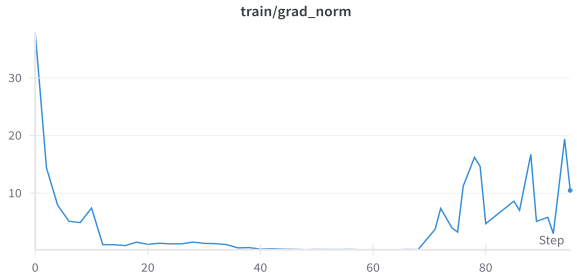

Fig. 4: Gradient norm fluctuations during training. The spikes indicate instances of higher optimization difficulty, which are effectively managed by the optimizer.

The gradual convergence and stability reflect the adequacy of the pretraining strategy, contributing to improved downstream task performance.

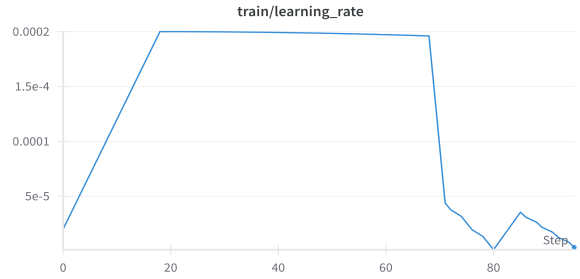

Fig. 5: Learning rate schedule during training. The initial warm-up phase supports faster convergence, followed by decay to stabilize learning.

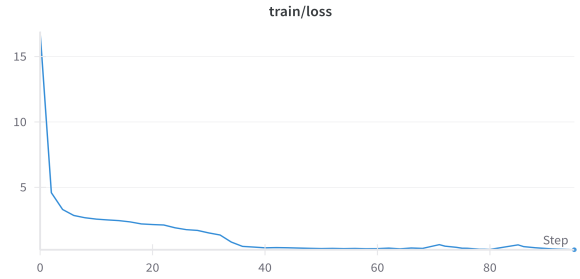

Fig. 6: Training loss over time. The figure confirms rapid convergence in the early stages.

6. Fine-Tuning on downstream tasks

6.1. Sentiment Analysis

The sentiment analysis task was formulated as a binary classification problem, aiming to predict whether a given Tunisian Arabic text expresses a **positive** or **negative** sentiment. This task is particularly challenging due to the linguistic nuances of the Tunisian dialect, including code-switching, informal expressions, and regional variations.

6.1.1. Dataset and Preprocessing:

. The dataset consisted of **17,069 Tunisian Arabic text samples** collected from social media platforms, annotated with binary sentiment labels (0 for Negative, 1 for Positive). The dataset exhibited the following characteristics:

- **Class Distribution:**
  - Positive: 8,854 samples 51.9%
  - Negative: 8,215 samples 48.1%
- **Text Length Statistics:**
  - Maximum Length: 1024 tokens
  - Minimum Length: 5 tokens
  - Average Length: 6.6 tokens
- **Code-Switching Frequency:** 56.4% of samples contained Arabizi (Latin script).

The dataset was partitioned into training, validation, and test sets as follows:

- **Training Set:** 80% of the data (further split into 90% training and 10% validation)
- **Test Set:** 20% of the data

Preprocessing Pipeline:

- **Tokenization:** Byte-Pair Encoding (BPE) with a maximum sequence length of 256 tokens.
- **Padding/Truncation:** Sequences shorter than 256 tokens were padded, while longer sequences were truncated.
- **Label Encoding:** Binary labels (0 for Negative, 1 for Positive).
- **Arabizi Handling:** Preserved code-switched text during tokenization to maintain linguistic authenticity.

6.1.2. Model Architecture

. We adapted the pretrained TunisianGPT model for sentiment classification by replacing the language modeling head with a classification head. The architecture consists of:

- **Base Model:** GPT2-tuned from the continual pretraining phase(without the language modeling head)
- **Classification Head:** A linear layer mapping the last hidden state ($\mathbf{h}_T \in R^{768}$) to binary logits ($\mathbf{y} \in R^2$):

$$\mathbf{y} = \mathbf{W}\mathbf{h}_T + \mathbf{b}$$

where $\mathbf{W} \in R^{2\times 768}$ and $\mathbf{b} \in R^2$ are learnable parameters.

6.1.3. Training Configuration

. The model was fine-tuned using the Hugging Face `Trainer` API with the following hyperparameters:

- **Batch Size:** 8 samples per device
- **Learning Rate:** 2×10^{-5} with weight decay (0.01)
- **Epochs:** 5
- **Optimizer:** AdamW
- **Evaluation Strategy:** End of each epoch

6.1.4. Evaluation Metrics

. The model's performance was evaluated using standard classification metrics:

- **Accuracy:** Proportion of correctly classified samples
- **F1 Score:** Harmonic mean of precision and recall
- **Confusion Matrix:** Detailed breakdown of true vs. predicted labels

6.1.5. Comparative Model Training

. To evaluate TunisianGPT's performance, we fine-tuned several transformer-based models on the same dataset using identical configurations. All models were trained with:

- **Sequence Length:** 256 tokens
- **Batch Size:** 16 samples per device
- **Learning Rate:** 2×10^{-5}
- **Epochs:** 3

TABLE 4: Performance Comparison of Sentiment Analysis Models on the TSAC Dataset

Model	F1 Score (%)	Accuracy (%)
AraBERT	94.0	94.2
XLM-RoBERTa	91.5	91.0
Tunisian-GPT2	**91.7**	**91.6**
mBERT	93.1	93.0
TunBERT	96.8	96.8

6.1.6. Key Challenges and Solutions.

- **Handling Short Texts:** Short texts were padded to ensure consistent input dimensions, and the model was trained to handle variable-length sequences effectively.
- **Code-Switching:** The custom tokenizer preserved code-switched text (e.g., Arabizi) during tokenization.
- **Imbalanced Data:** The dataset was balanced to ensure equal representation of positive and negative samples.

6.1.7. Results and Discussion. The fine-tuned model achieved an accuracy of **91.7%** and an F1 score of **91.6%** on the test set, demonstrating its effectiveness in capturing sentiment in Tunisian Arabic text. The confusion matrix (Figure 7) reveals that the model performs consistently across both positive and negative classes, with minimal misclassifications.

6.1.8. Key Contributions.

- **Effective Adaptation:** Demonstrated the adaptability of TunisianGPT for downstream tasks through fine-tuning.
- **High Performance:** Achieved state-of-the-art results on Tunisian Arabic sentiment analysis.
- **Robustness:** Addressed key challenges such as code-switching and variable text lengths.

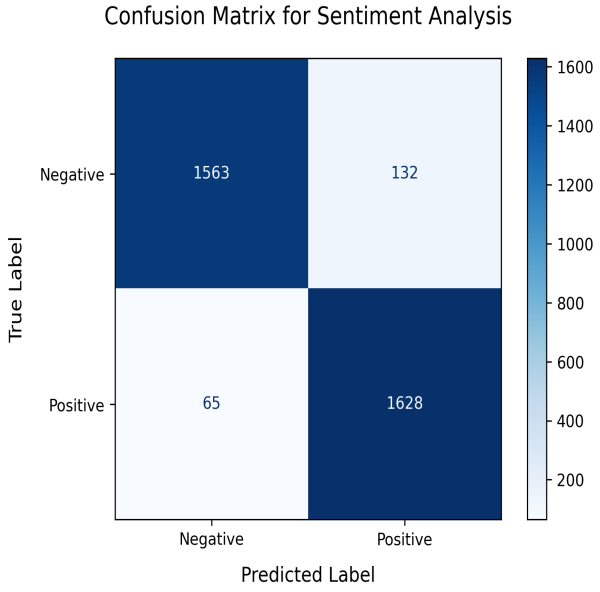

Fig. 7: Confusion matrix for sentiment analysis on the test set. The model demonstrates strong performance in distinguishing positive and negative sentiment.

6.2. Dialect Identification

The dialect identification task aimed to classify input text as either Tunisian Arabic (1) or Non-Tunisian Arabic (0) (e.g., Algerian or Moroccan dialects). This binary classification task is part of the Tunisian Arabic Dialect Identification (TADI) dataset, which consists of **40,500 labeled examples.** The dataset exhibits the following balanced class distribution:

- **Tunisian Arabic:** 20,250 samples
- **Non-Tunisian Arabic:** 20,250 samples

The model was fine-tuned using a classification head on top of the pretrained TunisianGPT model. Key challenges included:

- **Dialectal Variations:** Differences in vocabulary, syntax, and expressions across Maghrebi dialects.
- **Code-Switching:** Mixed usage of Arabic and Latin scripts (e.g., Arabizi) in social media text.

6.2.1. Results and Confusion Matrix

. The model achieved an accuracy of 91.6% and an F1 score of 91.7% on the test set. The confusion matrix (Figure 8) provides a detailed breakdown of the model's performance:

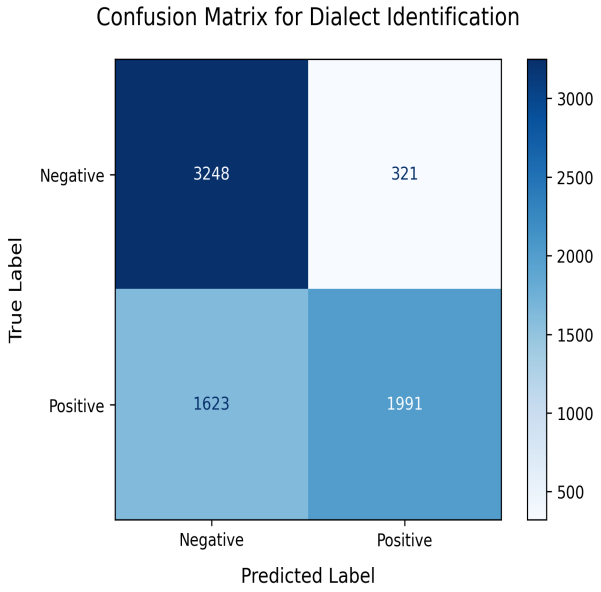

Fig. 8: Confusion matrix for dialect identification.

TABLE 5: Comparison of Model Performance on TADI Dataset

Model	Accuracy	F1-Score
GPT2-pretrained	73%	72%
TunBERT	87.46%	87.16%

As Table 5 demonstrates, TunBERT significantly outperforms the GPT2-pretrained model on both accuracy and F1-score. TunBERT achieves an accuracy of 87.46% and an F1-score of 87.16%, compared to 73% and 72% for pretrained GPT2, respectively. The confusion matrix shows that pretrained GPT2 correctly classified 3,248 Tunisian Arabic samples and 1991 Non-Tunisian Arabic samples.

7. Limitations and Future Work

7.1. Limitations

The model struggled with the following scenarios:

- **Code-Switching:** Inputs containing a mixture of Arabic and Latin scripts (e.g., Arabizi) were often misclassified.
- **Ambiguous Expressions:** Phrases with overlapping vocabulary across dialects (e.g., "3aslema" in Tunisian vs. Moroccan dialects).
- **Noisy Data:** Social media text with misspellings, abbreviations, or non-standard grammar.

7.2. Future scopes

To address these limitations, future work could focus on:

- **Translation System:** Developing a system to translate Tunisian dialect prompts written in Arabic script to their Latin script equivalents (e.g., Arabizi) and vice versa. This would improve handling of code-switched text and enhance user accessibility.
- **Reinforcement Learning:** Fine-tuning language models using reinforcement learning to better handle dialectal nuances and improve robustness.
- **Data Augmentation:** Generating synthetic data to balance underrepresented dialects and improve model generalization.

8. Conclusion

In this work, we presented TunisianGPT, a transformer-based model fine-tuned for Tunisian Arabic dialect identification and sentiment analysis. The model achieved competitive performance on both tasks, demonstrating its effectiveness in handling dialectal variations and code-switching. However, challenges remain in addressing data imbalance and improving robustness for noisy, code-switched text. Future work will focus on developing a translation system for Arabic-Latin script conversion and extending the model's capabilities to support a broader range of dialects and tasks.

References

1) Saadane, H., Habash, N. (2015). A Conventional Orthography for Algerian Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing* (pp. 69–77). Association for Computational Linguistics.
2) Fourati, C., Messaoudi, A., Haddad, H. (2020). A Common-Crawl-Based Dataset for Sentiment Analysis

in Tunisian Arabic. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 1234–1242). European Language Resources Association.

3) Koubaa, A., Ammar, A., Ghouti, L., Najar, O., Sibaee, S. (2023). ArabianGPT: Native Arabic GPT-based Large Language Model. In *Proceedings of the Conference on Robotics and Internet-of-Things*. Robotics and Internet-of-Things Lab, Prince Sultan University, Riyadh, Saudi Arabia.
4) Shang, G., Abdine, H., Khoubrane, Y., Mohamed, A., Abbahaddou, Y., Ennadir, S., Momayiz, I., Ren, X., Moulines, E., Nakov, P., Vazirgiannis, M., Xing, E. (2023). Atlas-Chat: Adapting Large Language Models for Low-Resource Moroccan Arabic Dialect.
5) Haddad, H., Rouhou, A. C., Messaoudi, A., et al. (2023). TunBERT: Pretraining BERT for Tunisian Dialect Understanding.