Arabic Named Entity Recognition (ANER)

1. Project Overview

- **Objective**: Automatically identify named entities (e.g., people, locations) in Arabic text.
- **Significance**: Tackles core challenges in Arabic NLP such as diacritics, tokenization, and morphological complexity.
- **Key Innovation**: Fine-tuning a BERT-based Arabic model with domain-specific enhancements for NER.

---

2. Dataset & Preprocessing

- **Dataset**: WikiFANE_Gold_2014_500K.txt
  - Size: ~15,763 sentences (~500,000 tokens)
  - Unique Tags: 102 (e.g., Nation, Population-Center, Politician, etc.)
- **Preprocessing**:
  - Diacritic removal, character normalization
  - Token masking for data augmentation

- Train/Test split: 90% / 10%

---

3. Model Architecture

- **Base Model**: `asafaya/bert-base-arabic`
  - 111M parameters
  - Pretrained on large Arabic corpora
- **Fine-tuning**:
  - Token classification head added
  - Approximately finetuned with ½ million token from dataset (https://fsalotaibi.kau.edu.sa/Pages-Arabic-NE-Corpora.aspx)
  - Fine-tuninig with 111M parameters
  - Label alignment for subword tokens
- **Training Setup**:
  - Learning Rate: 2e-5
  - Batch Size: 8
  - Epochs: 3
  - Early Stopping: Patience = 2
  - Hardware: Google Colab (T4 GPU)

---

4. Results & Evaluation

- **Test Set Performance**:
  - Accuracy: ~95%
  - F1-score (micro): ~0.68
- **Entity-wise F1 Scores**:
  - Nation: 0.81
  - Population-Center: 0.66
  - Politician: 0.64
- **Challenges**:
  - Lower recall for rare entities

- Entity boundary ambiguity in long/complex sentences

---

5. Demo & Inference

- **Example Input**:

  `"وُلِدَ محمد بن سلمان في الرياض عاصمة المملكة العربية السعودية."`

- **Output**:
  - "محمد بن سلمان" → Politician (96%)
  - "الرياض" → Population-Center (98%)
  - "المملكة العربية السعودية" → Nation (94%)
- **Interface**:
  - Gradio web UI for user interaction
  - Tkinter GUI for local desktop use

---

6. Limitations

1. **Sequence Length**: Truncated context due to BERT's 128-token limit
2. **Arabic-only Model**: Because we fine-tune on arabic dataset
3. **Overfitting**: Slight generalization drop after Epoch 1 (So we take model from epoch 1)
4. **Context Window**: Long sentences may lose coherence
5. **Vocabulary Mismatch**: OOV or dialectal words hurt performance
6. **Latency & Model Size**:
   - Large models can be slow or memory-intensive
   - Can be addressed using **Knowledge Distillation** or **Model Quantization**

---

7. Conclusion

- Successfully fine-tuned a BERT model for Arabic NER
- Achieved competitive accuracy and recall across common entity types
- Built interactive demos (Gradio & Tkinter)
- Open-sourced the model and codebase for reproducibility