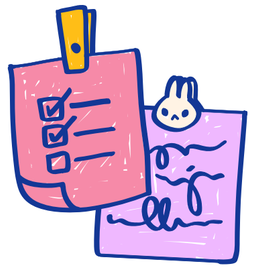
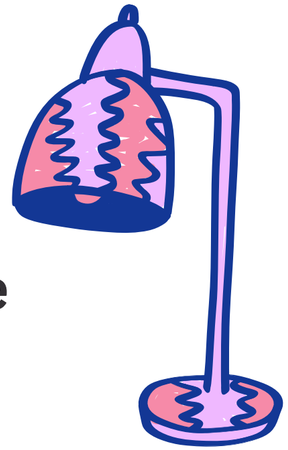
ARABIC AUTOCOMPLETE SYSTEM

Natural Language Processing Project

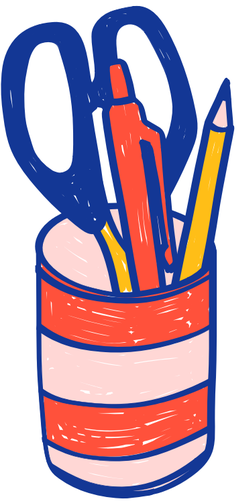
Content :-



Project Over view



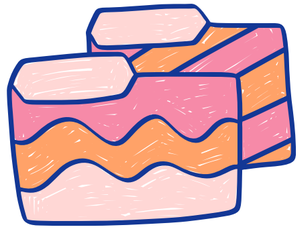
This project focuses on developing an intelligent Arabic autocomplete system that predicts the next word in a sentence based on user input, aiming to enhance typing efficiency and support Arabic language users. Given Arabic’s complex morphology and limited availability of high-quality autocomplete tools, this project leverages the Sanad Arabic text dataset to train an n-gram-based language model capable of understanding and completing user queries. The approach involves preprocessing and tokenizing Arabic text, constructing unigram to trigram models, and optimizing the system to provide real-time, accurate word suggestions.



Tools and Libraries Used:-

- **Programming Language:** Python
- **Libraries**:
  - Transformers: Hugging Face for model and tokenizer.
  - PyTorch: Model training and inference.
  - NumPy: Numerical computations.
  - NLTK: BLEU score calculation.
- **Environment:** Google Colab.
- **Model:** AraGPT2 (pre-trained, fine-tuned).

Dataset(SANAD):-



- **Source:** SANAD (Single-label Arabic News Articles Dataset).
- **Description:** Collection of Arabic news articles, suitable for language modeling contains seven categories [Culture, Finance, Medical, Politics, Religion, Sports and Tech], SANAD contains a total number of 190k+ articles.
- **Total number of articles:** 190k+ articles.

Preprocessing :-

- **Text Cleaning**:
  - Remove URLs and non-Arabic characters.
  - Normalize whitespace.
- **Arabic Normalization:**
  - Standardize alef variants.
  - Remove diacritics (e.g., fatha, kasra)
- **Sentence Validation:**
  - Ensure sentences are 3–50 words long.
  - Exclude sentences with only punctuation or low word diversity.
- **Tokenization**: AraGPT2 tokenizer for input IDs and attention masks.

Model Architecture :-

- **Base Model**: AraGPT2 (Arabic GPT-2 variant).
- **Train and test subset:**
  - Train : 20% randomly choosed from each category.
  - Test : 5% randomly choosed from each category.

Training Procedures :-

- **Training Arguments:**
  - ***num_train_epochs=2*** **: Model trains over the dataset twice.**
  - **per_device_train_batch_size=2 with gradient_accumulation_steps=4: Effective batch size = 8.**
  - **save_steps=1000: Save checkpoints every 1000 steps.**
  - **save_total_limit=3: Keeps only the latest 3 checkpoints.**
  - **fp16=True: Enables faster training using half-precision.**
  - **learning_rate=2e-5: Optimized for fine-tuning stability.**

Results :

- **Evaluation Metrics (2,660 test samples):**
  - ***Word-Level Accuracy: 31.24% (exact match).***
  - **Top-3 Accuracy: 44.89% (target in top-3 suggestions).**
  - **Character-Level Accuracy: 38.42%.**
  - **Average BLEU Score: 0.3232 (n-gram similarity).**

Further Improvements :

- **Extended Training**:
  - Increase epochs beyond 2 to reduce training loss (from 7.997).
  - Use learning rate scheduling to stabilize convergence.

- **Data Augmentation**:
  - Incorporate diverse Arabic corpora (e.g., social media, literature) to improve robustness.
  - Generate synthetic prompt-target pairs for underrepresented contexts.

THANK YOU