Darija Word Embedding Model

Project Overview

This project involved training a Word2Vec model using the Skip-gram architecture with negative sampling on a large Moroccan Darija corpus. The main objective was to generate quality word embeddings that capture the semantic and contextual relationships within Darija texts.

Tools & Environment

- Model: Word2Vec (Skip-gram, negative sampling)
- Library: Gensim
- Tokenizer: Custom regex-based tokenizer (removes punctuation, lowercases, splits on spaces)
- Platform: Local environment using Jupyter Notebook in VS Code
- Environment: Python 3.10 in a virtual environment named `darija_env`

System Challenges & Hardware Setup

We tested model training on two different computers, each with the same amount of RAM but differing in performance due to hardware components:

1. Maria’s Computer (Did Not Work)

- Specs: 8GB RAM, HDD, 4 cores
- Issues:
  - Even loading or training on just 1 million sentences caused CPU to freeze or stay idle.
  - The memory reached 94–97%, and Python dropped to 0% CPU during training, indicating a stall.
  - Disk read speeds from the HDD were also too slow to support large-scale training.

2. Meriem’s Computer (Successful Training)

- Specs: 32GB RAM, SSD (RAID), 8 cores
- Results:
  - 1 million sentences trained successfully in ~25 minutes
  - 3 million sentences trained in ~1 hour and 30 minutes
  - Training remained stable despite high memory usage (~94%) and 100% CPU during peak load
  - Gensim's multi-threaded model ran efficiently on this setup

Reflections

Attempting to train on the full 8.7 million sentence corpus was not feasible on either machine due to RAM constraints.

- Reducing to 3 million sentences produced high-quality embeddings without crashing.
- Using an SSD made a significant difference in performance and stability.
- With better hardware (e.g., cloud or university GPU VM), training could scale to full corpus and larger vector sizes.