Arabic dialects

UNSUPERVISED_PROJECT_REPORT

Norah Alqahtani, Batoul Alosaimi, and Shroaq Almutairi

Introduction

- Based on our belief in the importance of the Arabic language, we have worked on a project that serves the Arabic language and its dialects by making use of artificial intelligence and machine learning, Our Arabian countries consist of a lot of dialects and in order to classify texts into its original dialects we have created "Lahjatna" project that aims to Identify dialect of speech from Twitter tweets to one of the major arabic dialects (Nilotic, Gulf, Levantine, or Moroccan)

Objective:

- The objective of Lahjatna project is to help organizations and institutions recognize and identify their customers types and get to know them more in order to improve their services.

Design And Data Description:

- We worked on a public dataset from Kaggle
- https://www.kaggle.com/ahmedessam21/arabic-dialect-identificationfreelancing/version/2?select=train.tsv
- It contains 62,000 Arabic tweets, but it was not ready to use, during preprocessing we have cleaned data by removing “tashkeel”, removed repeated letters, correct spelling, simplify some writing ways, and remove stop words, etc.

Methodology:

- Load data
- Preprocessing
- Vectorization
- Topic modeling
- Label Tweet
- Exploratory_Data Analysis
- Prepare data for modeling
- Classification

Related works:

- DziriBERT: a Pre-trained Language Model for the Algerian Dialect:
    - they study the Algerian dialect which has several specificities that make the use of Arabic or multilingual models inappropriate. To address this issue, we collected more than one Million Algerian tweets, and pre-trained the first Algerian language model: DziriBERT. When compared to existing models, DziriBERT achieves the best results on two Algerian downstream datasets.
- The Arabic Dialect Identification for 17 countries (ADI17) Dataset:
    - This Research worked on classifying dialects YouTube content.

Algorithms:

- Building an unsupervised learning model including dimensionality reduction/topic modeling and/or clustering in python is required. These methods may be integrated into a complete recommendation system. Methods should be carefully selected via a combination of use case analysis and empirical feedback (e.g., quality of topics produced)

- Unsupervised modeling methods beyond those covered in the course are optional
- Supervised modeling methods (not covered in the course) are also optional, and should not replace unsupervised methods as the primary focus of the project (consult with an instructor if in doubt)

Tools:

- Python text processing libraries/tools (such as NLTK, spaCy, gensim, scikit-learn) are required for data handling.

- Other tools not covered in the course are optional but welcome
  - *Acquisition* tools could include web scraping libraries or use of APIs
  - *Storage* tools could include SQL or NoSQL (e.g., MongoDB) databases
  - *Processing* tools could include Google Cloud or Amazon Web Services for cloud computing resources
  - *Visualization* tools could include python libraries such as Bokeh and Plotly or resources outside of python such as Tableau
  - *Production* tools could include Flask or other web app libraries/technologies

Communication:

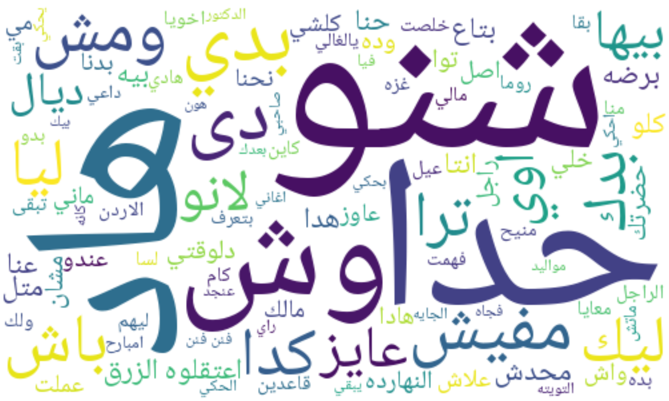


Figure 1: Word cloud for most common words in dataset

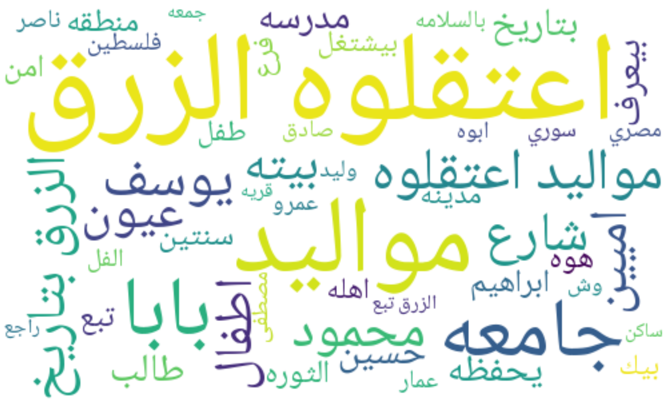


Figure 2: Word cloud for most common words in Nilotic dialect

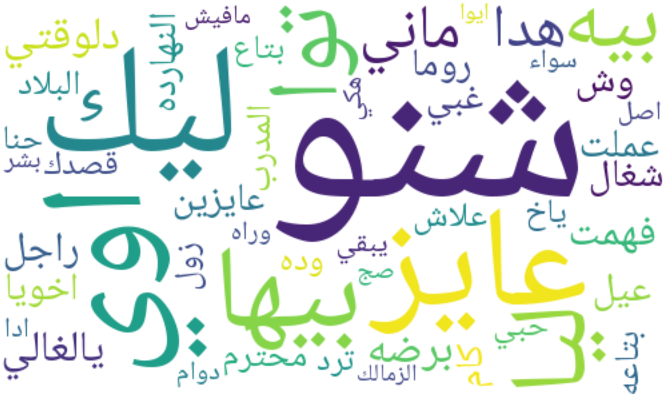


Figure 3:
Word cloud for most common words in Gulf dialect

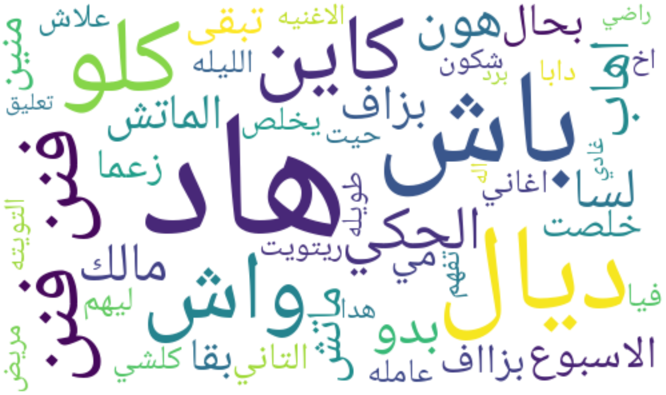


Figure 4:
Word cloud for most common words in Moroccan dialect

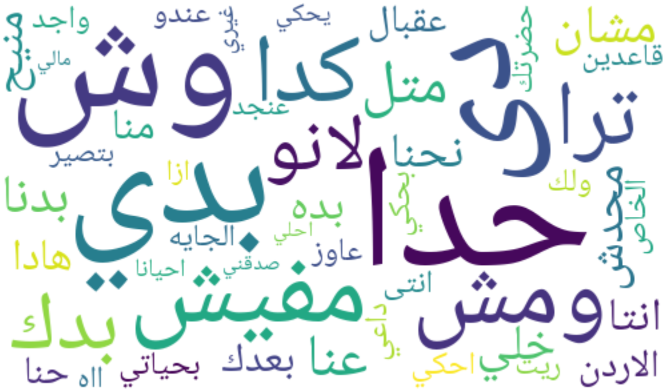


Figure 5
Word cloud for most common words in Levantine dialect

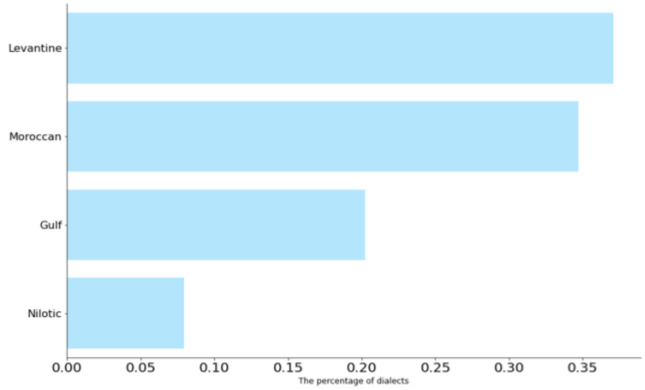
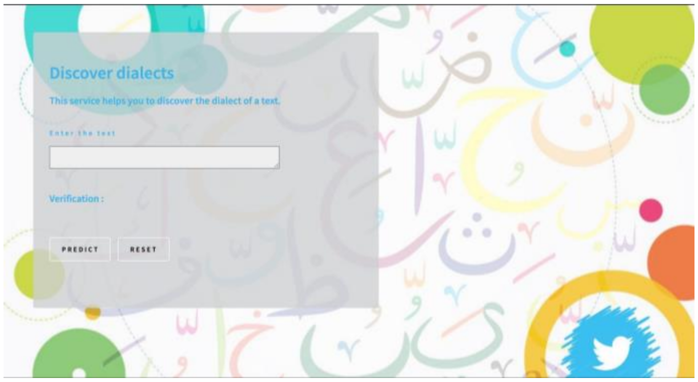


Figure 6:
Bar chart describing the distribution of dialects in the data



Discover dialects

This service helps you to discover the dialect of a text.

Figure 7:
The User Interface of our little site

Conclusion:

- We analyzed more than 62,000 tweets
- We created a model that able to analyze Arabic texts and predict its dialect
- We built a classification model with accuracy 0.75

future work:

- improve the results
- fetch our own data via scraping and gather twitter's data