Introduction

- **Sentiment Analysis** is a common task in text classification.
- **Sentiment Analysis** would help in several areas, such as politics and marketing.
- Twitter is a large microblogging website that handles a huge amount of data everyday, and analyzing these data would help to understand the opinions of these users.

Objectives

- To discuss several strategies for classifying, and analyzing sentiment of non English tweets.
- To use the Arabic language as a case study for applying the presented strategies, which include:
  - Data collection and annotation.
  - Data cleaning and preprocessing.
  - Calculating sentiment score for each sentence based on a sentiment lexicon.
  - Build and train the classification model using both the annotated dataset and the calculated sentiment score.

Dataset

A collected dataset by Nabil et al [1] was used that consists of 10K tweets:
- 6691 Objective
- 799 Positive
- 1684 Negative
- 832 Neutral

Methods

- **The dataset went through two stages:**
  - The data cleaning and preprocessing steps:
    - **Tweets:**
      - Remove characters and emojis
      - Extract punctuation and symbols
      - Normalize Arabic words
      - Remove non-Arabic characters
      - Analyze wrong words to extract correct words
      - Add sentiment score to each tweet
      - Cleaned Tweets + sentiment
  - **Two strategies were developed for the classification algorithm (Using SVM):**
    - Classifying as a single-stage process:
      - Randomly partitioning the dataset into 80% train and 20% test.
      - Build classification model (SVM)
      - Test Set
      - Tweet to SVM, POS, NGS or NEUTRAL
      - Accuracy for the single stage
    - Classifying as a two-stage process:
      - Convert set to Sub and Oog
      - First Stage
      - Build first classification model (SVM)
      - Second Stage
      - Build second classification model (SVM)
      - Test Set
      - Tweet to SVM, POS, NGS or NEUTRAL
      - Accuracy for the two stage

Results

- The measurements are calculated using the **accuracy**, and F1-score.
  \[
  \text{Accuracy} = \frac{T_{\text{Positive}} + T_{\text{Negative}}}{N}
  \]
  where \( N \) is the number of instances in the test set
  \[
  F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
  \]
  - The algorithms were run 100 times for each of the following experiments:
    - **Cleaned tweets without adding the sentiment score:**
      - **Measurement**
        - **Maximum**
        - **Minimum**
        - **Average**
      - **Accuracy**
        - 70.2
        - 67.8
        - 68.9
      - **F1-Score**
        - 62.01
        - 58.5
        - 60.3
    - **Cleaned tweets with adding sentiment score for each tweet:**
      - **Measurement**
        - **Maximum**
        - **Minimum**
        - **Average**
      - **Accuracy**
        - 74.5
        - 72.4
        - 73.3
      - **F1-Score**
        - 68.3
        - 65.2
        - 66.6
    - **Cleaned tweets with sentiment scores using two-stage approach:**
      - **Measurement**
        - **Maximum**
        - **Minimum**
        - **Average**
      - **Accuracy**
        - 73.3
        - 68.7
        - 70.7
      - **F1-Score**
        - 69.6
        - 64.03
        - 67.02

Discussion and Analysis

- A lexicon sentiment can help to improve the classification algorithm for the sentiment problem.
- Comparing a single-stage model and a two-stage model using a T test gives a P-value < 0.0001, which means a single-stage is statistically significant with a mean accuracy of 97.3%
- The confusion matrix of the second stage of the two-stage approach shows there is an overlap between natural, and negative classes.

Conclusion

- An approach for sentiment problem was presented and tested using the Arabic language.
- Three classification models were implemented and discussed.
- Future work: 1) testing our implemented models using another dataset in Arabic, 2) applying this approach to other languages.

References