ARABIC SENTIMENT ANALYSIS

Sarah Alsubhi

ABSTRACT

The goal of this study is
To determin whether Arabic tweets can classified either as displaying positive, or negative sentiment with considering emojies using different models

01.

INTRODUCTION

WHAT IS SENTIMENT ANALYSIS

- Sentiment analysis lets you analyze the sentiment behind a given piece of text.
- It combines machine learning and natural language processing (NLP) to achieve this.

WHAT IS SENTIMENT ANALYSIS

- Sentiment analysis is a powerful technique in Artificial intelligence that has important business applications.

- e.g. to understand your customers' attitude towards your product.

02.

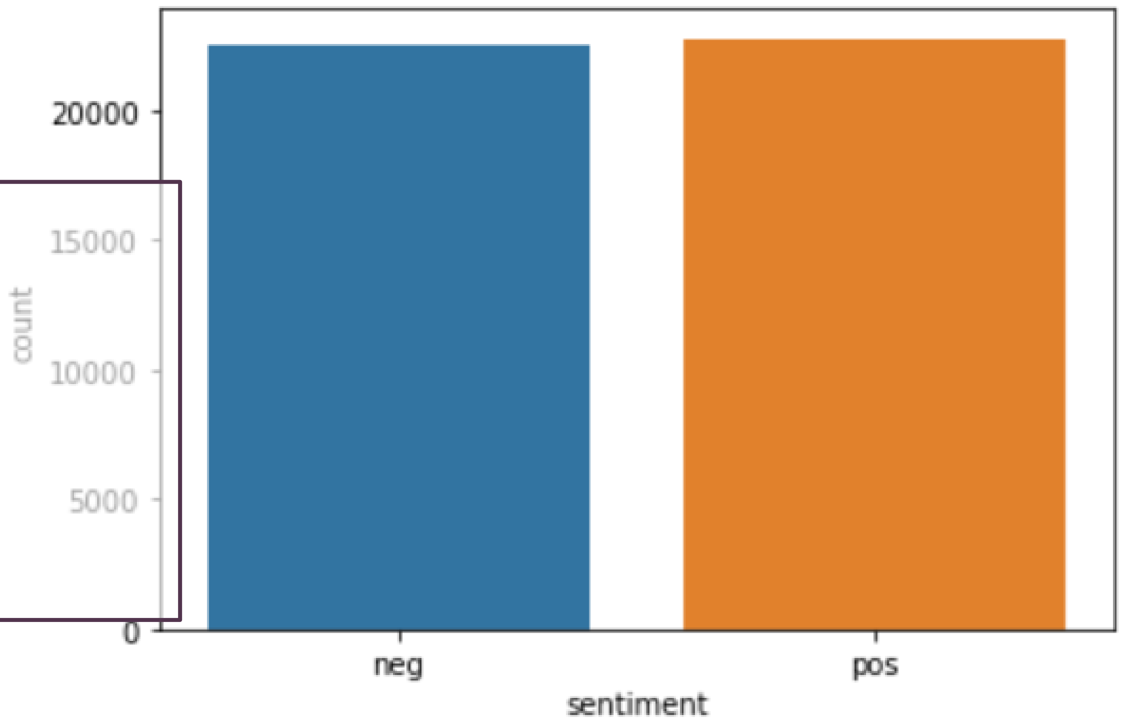
DATASET

ARABIC SENTIMENT TWITTER CORPUS

58,000

This dataset we collected in April 2019. It contains 58K Arabic tweets (47K training, 11K test) tweets annotated in positive and negative labels. The dataset is balanced and collected using positive and negative emojis lexicon.

BALANCED DATASET



03. METHODOLOGY

3.1 PREPROCCESSING

CLEANING THE DATA:

- remove punctuations, longation
- remove URL, usernames
- remove special characters, Numbers
- remove Tashkeel
- Remove stopwords
- Tokenization and Stemming

	sentiment	tweets	Tokenized	stemmed
0	neg	اعترف ان بتس كانو شوي شوي يجيبو راسي اليوم بال...	[اعترف, ان, بتس, كانو, شوي, شوي, يجيبو, راسي, ...	[ا, ع, ت, ر, ف, , ا, ن, , ب, ت, س, , ك, ا, ...
1	neg	توقعت اذا جات داريا بشوفهم كاملين للحين احس اح...	[توقعت, اذا, جات, داريا, بشوفهم, كاملين, للحين...	[ت, و, ق, ع, ت, , ا, ذ, ا, , ج, ا, ت, , د, ...
2	neg	الاهليالهلال اكتب توقعك لنتيجه لقاء الهلال وال...	[الاهليالهلال, اكتب, توقعك, لنتيجه, لقاء, الهل...	[ا, ل, ا, ه, ل, ي, ا, ل, ه, ل, ا, ل, , ا, ك, ...
3	neg	نعمه المضادات الحيويه تضع قطره💧مضاد بنسلين بكت...	[نعمه, المضادات, الحيويه, تضع, قطره, 💧, مضاد, ...	[ن, ع, م, ه, , ا, ل, م, ض, ا, د, ا, ت, , ا, ...
4	neg	الدودو جايه تكمل علي 💔	[الدودو, جايه, تكمل, علي, 💔]	[ا, ل, د, و, د, و, , ج, ا, ي, ه, , ت, ك, م, ...

3.1 PREPROCCESSING

BOW:

- Get dataset bag-of-words counts as a vector using **COUNTVECTORIZER**

```
from sklearn.feature_extraction.text import CountVectorizer
#get dataset bag-of-words counts as a vector
bow_transformer = CountVectorizer(analyzer=tokenize).fit(df['tweets'])
```

```
# BoW vector representation
messages_bow = bow_transformer.transform(df['tweets'])
```

3.2 FEATURES

TEXTUAL:

- Length of tweets
- Number of words count
- Number of characters
- Number of sentences
- Average words length
- Average sentence length.

Tweets_len	word_count	char_count	avg_word_length	avg_sentence_lenght
11	11	76	6.909091	11.0
12	12	87	7.250000	12.0
15	15	127	8.466667	15.0

3.2 FEATURES

WORD EMBEDDING:

- TfidfVectorizer on tweets

```
# make pipeline
pipe = make_pipeline(TfidfVectorizer(),
```

- TfidfVectorizer on BoW

```
#transform the entire bag-of-words corpus into TF-IDF corpus
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer().fit(messages_bow)
```

3.2 FEATURES

WORD EMBEDDING:

- Word2Vec

```
model = gensim.models.Word2Vec(sentences=tweets_lines,
                               vector_size = embedding_dim, window=5,
                               min_count=1) #list of sentances (tokens)
words = list(model.wv.index_to_key) #vocab size
```

- Testing Word2Vec

```
sample = model.wv["حسن"]
print(sample.shape)
#print(sample)
print(model.wv.most_similar("حسن"))
```

```
(,100)
0.993261158466 ,'فكن') ,(0.9943529367446899 ,'امري') ,(0.995944619178772 ,'اليقين')]
0.9923 ,'عذاب') ,(0.9929375052452087 ,'فرحه') ,(0.9930124878883362 ,'صدر') ,(3391
بالمذ') ,(0.9917353987693787 ,'راحه') ,(0.9918506741523743 ,'ياارب') ,(112988471985
[(0.9913020133972168 ,'الصلاه') ,(0.9914056658744812 ,'ل
```

3.3 MODELS

MACHINE LEARNING MODELS:

- LogisticRegression( )

```
lr_ = LogisticRegression()
lr_.fit(X_train,y_train)
prediction = lr_.predict(X_test)
print(classification_report(y_test, prediction))
```

- MultinomialNB( )

```
sentiment_model_ = MultinomialNB().fit(X_train, y_train)
prediction = sentiment_model_.predict(X_test)
print(classification_report(y_test, prediction))
```

3.3 MODELS

MACHINE LEARNING MODELS:

- Word2Vec + LSTM

```
model = Sequential()

embedding_layer = Embedding(num_words,
                            embedding_dim,
                            embeddings_initializer=Constant(embedding_matrix),
                            input_length=max_length,
                            trainable=False)

model.add(embedding_layer)
model.add(LSTM(units=32, dropout=0.2, recurrent_dropout=0.2 )) #dropout to deact
model.add(Dense(1,activation='sigmoid'))
```

3.3 MODELS

MACHINE LEARNING MODELS:

- Bi-LSTM

```
model.add(layers.Embedding(input_dim=500,
                           output_dim=100,
                           input_length=X.shape[1]))
model.add(layers.Bidirectional(layers.LSTM(100, dropout=0.5,
                                           recurrent_dropout=0.5,
                                           return_sequences=True)))
model.add(layers.GlobalMaxPool1D())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(dropout))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dropout(dropout))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dropout(dropout))
model.add(layers.Dense(1, activation='sigmoid'))
```

04.

RESULTS

ML MODELS

		LR-TFIDF			LR-TEXT			LR-BOW-TFIDF			NB-BOW-TFIDF		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
CLASS	Pos	71	82	76	53	48	50	96	81	88	87	79	83
	Neg	79	67	73	53	58	55	84	96	90	81	88	84
ACC		75			53			89			84		

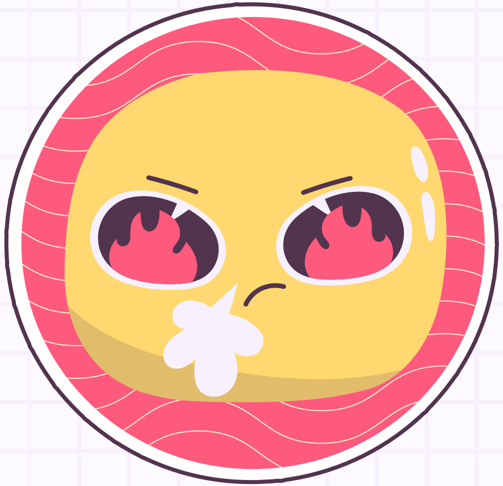
DL MODELS

	W2V+LSTM		BI-LSTM	
EVAL	LOSS	ACC	LOSS	ACC
TRAIN	60	63	23	87
TEST	60	63	22	87

CHALLENG

Dealing with Arabic text

- Wordembedding :Aravec, word 2vec
- resources



NEXT

Improving the models
Experiment new models

THANKS!