

Sarcasm Detection

Problem Description

Humans have a social nature. Social nature means that we interact with each other in positive, friendly ways, and it also means that we know how to manipulate others in a very negative way. Sarcasm, which is both positively funny and negatively nasty, plays an important part in human social interaction. We will build a machine learning algorithm to detect word whether sarcastic or not on an Arabic and English dataset.

Arabic Model Design

Step 1: Collect Data

The dataset was created using previously available Arabic sentiment analysis datasets (SemEval 2017 and ASTD) and adds sarcasm and dialect labels to them. The dataset contains 10,547 tweets, 1,682 (16%) of which are sarcastic. For more details, please check our paper [From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset](#).

ArSarcasm is provided in a CSV format, we provide an 80/20 train/test split to keep things consistent for future comparisons. The training set contains 8,437 tweets, while the test set contains 2,110 tweets.

The dataset contains the following fields:

- tweet: the original tweet text surrounded by quotes ("").
- sarcasm: boolean the indicates whether a tweet is sarcastic or not.
- sentiment: the sentiment from the new annotation (positive, negative, neutral).
- original_sentiment: the sentiment in the original annotations (positive, negative, neutral).
- source: the original source of tweet SemEval or ASTD.
- dialect: the dialect used in the tweet, we used the 5 main regions in the Arab world, follows the labels and their meanings:
 - msa: modern standard Arabic.
 - egypt: the dialect of Egypt and Sudan.
 - levant: the Levantine dialect including Palestine, Jordan, Syria and Lebanon.
 - gulf: the Gulf countries including Saudi Arabia, UAE, Qatar, Bahrain, Yemen, Oman, Iraq and Kuwait.
 - magreb: the North African Arab countries including Algeria, Libya, Tunisia and Morocco .

Step 2: Prepare the data

As someone who is used to working with English texts, I found it difficult in the first place to translate preprocessing steps routinely used for English texts to Arabic. Luckily, I later came across a Github repository with the code for cleaning texts in Arabic. The steps basically involve removing punctuation, Arabic diacritics (short vowels and other harakahs), elongation, and stopwords (which is available in NLTK corpus).

Step 3: Choose the model

1. We have choose Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, every pair of features being classified is independent of each other.
2. calculate frequency of each word in positive and negative sentence
3. calculate likelihood and prior then get log of them by getting probability of features in positive and negative given probability of output
4. test new records by get getting probability of features in positive and negative then normalize it and make prededction depending on bigger probability (argmax)
5. calculate accuracy

Step 4: Evaluation

The steps involved in Arabic sentiment analysis using naive bayes . The major difference between Arabic and English NLP is the pre-processing step. All the classifiers fitted gave impressive accuracy score 96.04%

Step 5: Prediction

Testing on new data gives us satisfying results and the 83.06% for accuracy

English Model Design

Step 1: Collect Data

Past studies in Sarcasm Detection mostly make use of Twitter datasets collected using hashtag based supervision but such datasets are noisy in terms of labels and language. Furthermore, many tweets are replies to other tweets and detecting sarcasm in these requires the availability of contextual tweets.

To overcome the limitations related to noise in Twitter datasets, this News Headlines dataset version 2.0 for Sarcasm Detection is collected from two news website. TheOnion aims at producing sarcastic versions of current events and we collected all the headlines from News in Brief and News in Photos categories (which are sarcastic). We collect real (and non-sarcastic) news headlines from HuffPost.

Each record consists of three attributes:

- a. `is_sarcastic`: 1 if the record is sarcastic otherwise 0
- b. `headline`: the headline of the news article
- c. `article_link`: link to the original news article. Useful in collecting supplementary data

Step 2: Prepare the data

- 1- convert all words to lowercase
- 2- remove punctuation marks
- 3- remove digits
- 4- tokenize sentence
- 5- stemming results
- 6- remove stop words

Step 3: Choose the model

We have choose Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, every pair of features being classified is independent of each other.

- 1- calculate frequency of each word in positive and negative sentence
- 2- calculate likelihood and prior then get log of them by getting probability of features in positive and negative given probability of output
- 3- test new records by get getting probability of features in positive and negative then normalize it and make prededction depending on bigger probability (argmax)
- 4- calculate accuracy

Step 4: Evaluation

The steps involved in Arabic sentiment analysis using naive bayes. The major difference between Arabic and English NLP is the pre-processing step. All the classifiers fitted gave impressive accuracy scores 87.75% no better than arabic but sounds good

Step 5: Prediction

Testing on new data gives us satisfying results and the 79.89% for accuracy

Model performance

The same model using in two datasets(Arabic & English)with a difference in preprocessing the model achieve accuracy better in Arabic dataset =83.06 than English dataset =79.89 and that is very satisfying

Experimental Result

After fitting the train data and getting training accuracy we got a 79.89% for English and 83.07% for Arabic.