Twitter Sentiment Analysis

Daniele Gerbaldo,

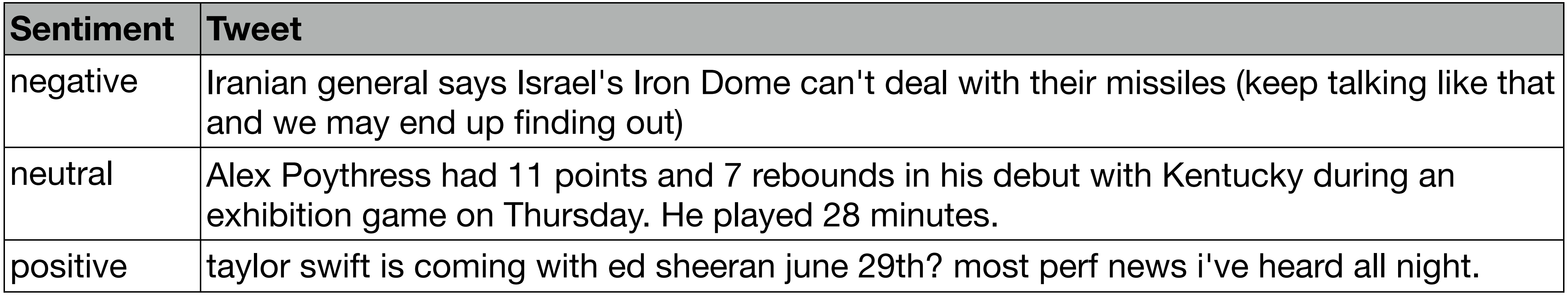
Thieu Nguyen, Tim Röthig

Agenda

1. Problem description
2. Dataset
3. Data preparation
4. Model Type selection
5. Model fitting
6. Model evaluation
7. Other Model Type

Problem Description

- Sentiment classification of Tweets
- Either negative, neutral or positive
- Data instances:



Sentiment	Tweet
negative	Iranian general says Israel's Iron Dome can't deal with their missiles (keep talking like that and we may end up finding out)
neutral	Alex Poythress had 11 points and 7 rebounds in his debut with Kentucky during an exhibition game on Thursday. He played 28 minutes.
positive	taylor swift is coming with ed sheeran june 29th? most perf news i've heard all night.

- Goal: Build a classifier, that classifies the tweets correctly

Dataset

- SemEval-2014 Task 9
- Normally 18.000 instances
- Several tweets unavailable
- Several tweets classified multiple times in different categories

➢5000 usable instances

➢Only performed the classification on two classes (positive and negative)

Used python libraries

- Pandas: Used for managing the data

- NLTK (Natural Language Tool Kit): Used for the natural language processing

- scikit-learn: Used for the model fitting and the error measurement

Data Preparation

- Document Term Matrix:
  - Features: Unique words of every Tweet
  - Feature values: Number the features that are occurring in the tweet

Sentiment	Tweet
positive	Breezin' won the Best Pop Instrumental Performance at the 19th Grammy Awards

Sentiment	won	best	pop	instrumental	performance	grammy	awards
positive	1	1	1	1	1	1	1

Data Preparation: Feature extraction

1. Remove links from every tweet
2. Merge all Tweets to one large string
3. Tokenize to retrieve all the words occurring in the tweets using the NLTK word_tokenize() function
4. Remove punctuation and numbers
5. Remove stop words using the build in NLTK stop words library
6. Stem the words down to there root form using the NLTK PorterStemmer() function
7. Remove the infrequent words that occur less then 5 times
8. Use the set() function of python to retrieve all the unique words

Data Preparation: Infrequent word removal

- Also removes usernames and unimportant hashtags
- Word frequency threshold is an important hyperparameter
  - ➢Needs to be tuned
- Testing different thresholds with the default SVM of scikit-learn

- word_frequency=3
  - ➢Test Accuracy = 75%
- word_frequency=5
  - ➢Test Accuracy = 79%

- word_frequency=7
  - ➢Test Accuracy = 77%
- word_frequency=10
  - ➢Test Accuracy = 76%

Data Preparation: Dataset creation

1. Repeat Step 1. - 5. from the previous slide for every tweet
2. Count how often each feature is occurring in the tweet

Sentiment	won	best	pop	tree	performance	fire	grammy	flute	apple	night	lost
positive	2	1	1	0	1	0	1	0	0	0	0
negative	0	0	0	1	0	1	0	0	0	2	1

3. Split the Dataset in a Training set (80%) and a Testing set (20%)

Model Type Selection

- Support Vector Machine (SVM)
  - Insensitive to the number of dimensions
  - First choice for text classification
- Multinomial Bayes
  - Fast
  - Robust to unnecessary features
  - Good baseline text classifier

SVM: Model fitting

- Tunable Hyperparameters were optimized using grid search:
    - C: The degree to which misclassification is punished
        - Tried Values = [0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 10]
    - Kernel: The used Kernel type
        - Tried Values = [‘linear’, ‘poly’, ‘rbf’]
    - Degree: The degree of the polynomial in case the kernel function is polynomial
        - Tried Values = [2, 3, 4, 5, 10]

SVM: Model Evaluation

- C=3 preformed best
  - Accuracy was decreasing for higher and lower values

- Polynomial Kernel function performed worst no mater which degree was chosen
  - Avg. Accuracy: 74%
- Linear Kernel function performed reasonable
  - Avg. Accuracy: 80%
- Radial Basis Kernel function performed best
  - Avg. Accuracy: 82%

SVM: Model Evaluation

- Kernel=‘linear‘, C=0.5
  - Test Accuracy = 78%
- Kernel=‘linear‘, C=1
  - Test Accuracy = 80%
- Kernel=‘linear‘, C=3
  - Test Accuracy = 82%
- Kernel=‘linear‘, C=5
  - Test Accuracy = 77%

- Kernel=‘rbf‘, C=0.5
  - Test Accuracy = 75%
- Kernel=‘rbf‘, C=1
  - Test Accuracy = 78%
- Kernel=‘rbf‘, C=3
  - Test Accuracy = 84%
- Kernel=‘rbf‘, C=5
  - Test Accuracy = 83%

SVM: Chosen Model

- Kernel = ’rbf’ (radial basis function)
- C = 3

- ➢Accuracy was determined by taking the average over 3 SVM trained on differently sampled datasets
  - ➢Train Accuracy = 95%
  - ➢Test Accuracy = 85%

Multinomial Bayes

- Another descent method for text classification
- alpha = 1

- Performed worse
    - Train Accuracy = 88%
    - Test Accuracy = 78%