U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Sentiment analysis in tweets

Natural Language Processing

Artificial Intelligence

2nd project - Final delivery

David Silva - up201705373
Luís Cunha - up201706736
Manuel Coutinho - up201704211

1. Problem specification

. Part of the E-c (emotion classification) task of SemEval-2018*.

. The task's goal is to classify each individual Tweet as **'neutral** or no emotion' or as **one, or more**, of **eleven given emotions** that best represent the mental state of the tweeter.

Example:

"I have the best girlfriend in the world 😍 #blessed"

Anger	Anticipation	Disgust	Fear	Joy	Love	Optimism	Pessimism	Sadness	Surprise	Trust
0	0	0	0	1	1	1	0	0	0	0

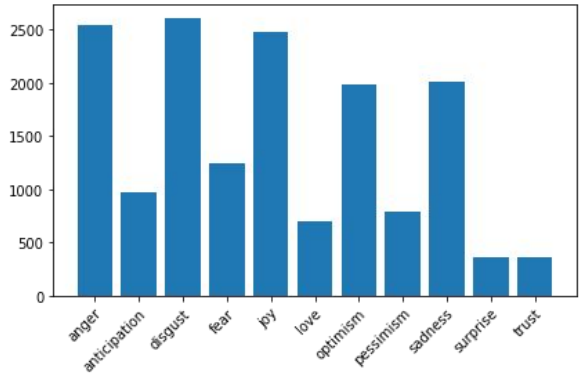
* SemEval-2018 Task 1: Affect in Tweets (AIT-2018)

2. Dataset analysis

. Tweet data contains a lot of **noise**. As is common in social media, text is **short**, **irregular** and makes use of special constructs such as **emoji**, **hashtags**, **slang** and **abbreviations**.

“ Only I could be talking to a catfish on tinder 🤦‍♀️😂 glad I don't use it srsly 🤣 “

. We observed that the distribution of **emotions** in the dataset is **unbalanced**:



* training dataset contains 6838 tweets

3. Pre-processing

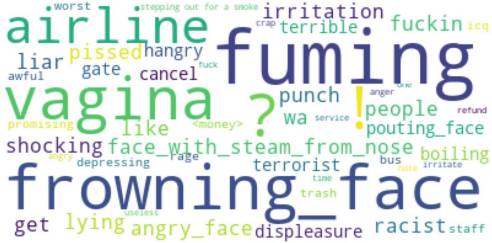
. **Tokenization** stage, used the **ekphrasis** social network tokenization library:

- **Emoji** to meaning (‘😂’ to ‘laugh’)
- Remove **mentions** (‘@POTUS’ to ‘<user>’)
- Split **hashtags** (‘#content’ to ‘<hashtag> content </hashtag>’)
- Normalize **letter repetition** (‘happyyyyy’ to ‘happy <elongated>’)
- Removing **stopwords**
- **Spell correction**
- Using **lemmatization** (or **stemming**) to reduce word space (‘better’ to ‘good’)
- **Slang translation** (‘tbh’ to ‘to be honest’)

. **Representing** a tweet using **TF-IDF** vector or **word-embeddings**

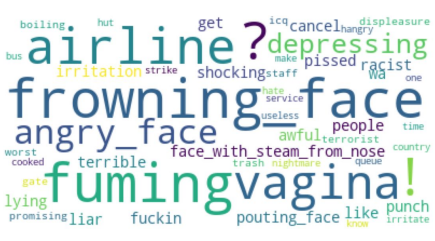
4. TF-IDF word-cloud

. Word cloud, by class, of the TF-IDF distribution:

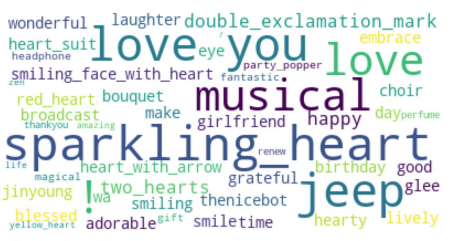


Anger

Anticipation

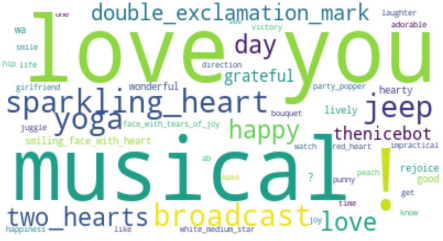


Disgust



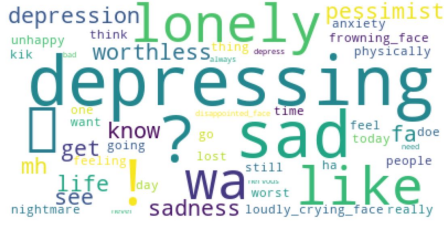
Joy

Fear

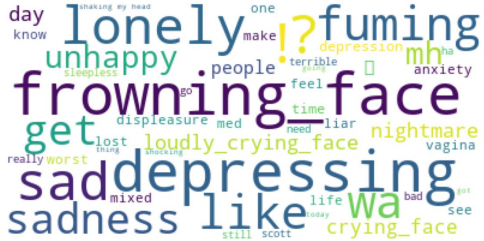


Love

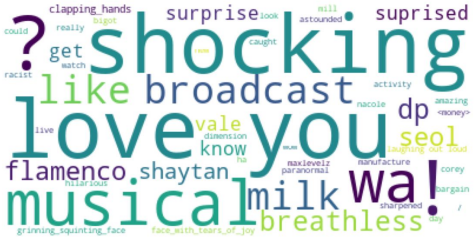
Optimism



Pessimism



Sadness



Surprise

Trust

5. Classification

. We solved the classification step by using **two approaches:** using the **TF-IDF** vectorization and **six** algorithms:

- Naïve-Bayes *baseline
- Logistic Regression
- Support Vector Classification
- Perceptron
- Decision Tree Classification
- Random Forest Classification

. Using **LSTM** neural networks with **word embeddings** representation

6. Results

. Evaluation of algorithms using **stemming**, the **default** algorithm **parameters** and all of the **tokenization transformations** except for the slang dictionary..

Algorithms	Jaccard score	F1(macro)	F1(micro)	Training Time (s)
Naïve-Bayes	0.203	0.173	0.318	1.177
Logistic Regression	0.480	0.531	0.609	2.439
Support Vector Class	0.476	0.461	0.606	80.146
Perceptron	0.409	0.463	0.544	1.211
Decision Tree Class.	0.369	0.424	0.508	11.045
Random Forest Class.	0.369	0.348	0.503	46.369

6. Results (cont.)

. Evaluation of the **optimized** algorithms (NB and Logistic Regression). The last row represents the results obtained by the **best team at SemEval-2018** (psyML).

Algorithm	Jaccard	F1 (macro)	F1 (micro)	Training Time
Naïve-Bayes	0.312	0.241	0.432	1,170
Logistic Reg.	0.508	0.537	0.636	6.999
Emb + LSTM	0.551	0.453	0.663	19.708 *
SemEval-2018 winner	0.588	0.528	0.701	-

* Deep learning model trained on a NVIDIA GTX 1060 GPU

6. Results (cont.)

. Confusion table and performance measures for **optimized Logistic Regression**:

	True-Neg.	False-Pos.	False-Neg.	True-Pos.	Precision.	Recall	F-measure	Support
Anger	0.55	0.10	0.11	0.24	0.71	0.69	0.70	315
Anticipation	0.72	0.14	0.08	0.06	0.29	0.40	0.33	124
Disgust	0.53	0.11	0.12	0.24	0.68	0.66	0.67	319
Fear	0.80	0.07	0.03	0.10	0.61	0.76	0.68	121
Joy	0.46	0.09	0.10	0.35	0.79	0.78	0.78	400
Love	0.75	0.10	0.05	0.10	0.50	0.65	0.56	132
Optimism	0.52	0.13	0.10	0.25	0.66	0.72	0.69	307
Pessimism	0.78	0.11	0.06	0.06	0.34	0.50	0.41	100
Sadness	0.59	0.11	0.12	0.18	0.62	0.60	0.61	265
Surprise	0.93	0.03	0.02	0.02	0.39	0.43	0.41	35
Trust	0.90	0.05	0.04	0.01	0.18	0.23	0.20	43

6. Results (cont.)

. Confusion table and performance measures for the solution using **embeddings + LSTM** neural net.:

	True-Neg.	False-Pos.	False-Neg.	True-Pos.	Precision.	Recall	F-measure	Support
Anger	0.57	0.08	0.09	0.27	0.78	0.76	0.77	315
Anticipation	0.86	0.00	0.14	0.00	1.00	0.01	0.02	124
Disgust	0.54	0.10	0.10	0.26	0.72	0.72	0.72	319
Fear	0.84	0.02	0.05	0.09	0.79	0.65	0.71	121
Joy	0.50	0.05	0.10	0.35	0.88	0.77	0.82	400
Love	0.83	0.02	0.09	0.05	0.73	0.36	0.48	132
Optimism	0.56	0.09	0.09	0.25	0.73	0.73	0.73	307
Pessimism	0.88	0.00	0.10	0.01	0.70	0.07	0.13	100
Sadness	0.66	0.04	0.16	0.14	0.79	0.48	0.60	265
Surprise	0.96	0.00	0.04	0.09	1.00	0.06	0.11	35
Trust	0.95	0.00	0.05	0.00	0.00	0.00	0.00	43

5. References and research

. NLP concepts:

- Lectures slides.
- Bird, S., et al. (2009). Natural Language Processing with Python.

. Preprocessing:

- Haddi E., et al. (2013). The Role of Text Pre-processing in Sentiment Analysis.
- Prakash N., Amalanathan A. (2019). Data preprocessing analysis using Twitter data.
- Zin H., et al. (2017). The effects of pre-processing strategies in sentiment analysis of online movie reviews
- Singh, T., Kumari, M. (2016). Role of Text Pre-processing in Twitter Sentiment Analysis.
- Angiani G, et al. (2016). A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter
- Tutorial on text preprocessing using NLTK and Scikit-learn: Sentiment analysis of reviews: Text Pre-processing
- Social network text processing: Ekphrasis- Social Network tokenization

. Algorithms:

- NTLK Naïve Bayes example: 6. Learning to Classify Text (Naïve Bayes)
- ScikitLearn SVM: 1.4. Support Vector Machines
- ScikitLearn K-Nearest Neighbours: 1.6. Nearest Neighbors
- Classification using neural networks: Practical Text Classification With Python and Keras