CLASSIFICATION PROJECT

The aim of this project is to analyze a Twitter collection about offensive language and hatred.

The collection is stored in the file “Hatred labelled dataset” made up of the following the fields:

- count = number of CrowdFlower users who coded each tweet.
- hate_speech = number of CF users who judged the tweet to be hate speech.
- offensive_language = number of CF users who judged the tweet to be offensive.
- neither = number of CF users who judged the tweet to be neither offensive nor non-offensive.
- class = class label for majority of CF users. 0 - hate speech 1 - offensive language 2 – neither.
- tweet: textual information to process.

Using the field “tweet”, it is necessary to follow the next steps:

1. Preprocessing

Mandatory preprocessing steps

- Remove unseful data: ! "_ $% & / ( ) = _ ˆ* ¡@
- Remove all capital letters
- Correct wrong words: https://norvig.com/spell-correct.html
- Lemmatize all terms

Optional preprocessing steps:

- Remove contractions: don’t → do not[1]
- Remove repeated words: great great show
- Replace emoticons, for example, 😊 with “smile”, 😉 with “ok”

2. Vectorization

Vectorize every tweet by following different configurations:

- TFIDF
- TFIDF + N-grams
- TFIDF + N-grams + POS tagging
- TFIDF + N-grams + POS tagging + other features: RTs, number of words, number of sentences, sentiments, hatred n-gram dictionary*

[1] https://englishstudypage.com/grammar/list-of-contractions-in-english/

*If any n-gram detected in a tweet is found in the file “hatred n-gram dictionary”, the corresponding associated weight should be included as a feature, otherwise 0.

3. Feature selection

Select the best features using the selectKBest and removing 70% of the features used per configuration.

4. Classification algorithm

Using the selected best features, use 2 classification algorithms to classify the tweets according to the field “class” (0 - hate speech 1 - offensive language 2 – neither). To do so, 70% of the dataset will be used for training and 30% for testing. Tune the different parameters, if possible, by a cross validation.

Write a report describing the process followed (use ACM template):

- Problem description
- Methods and materials: Classification algorithm and dataset for testing
- Experiments and results: containing the following evaluations metrics: precision, recall, F-measure and confusion matrices
- Conclusions