Hate Speech and Offensive Language

Text Classification and Clustering

Emanuela Elli (892901)
Federica Madon (825628)
Tommaso Strada (829351)

Text Mining and Search - 2022/2023

About the dataset

The dataset is about **tweets** that may contain *hate speech*, *offensive language* or *neither*. There are **24783 rows** and **6 columns**.

Columns

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

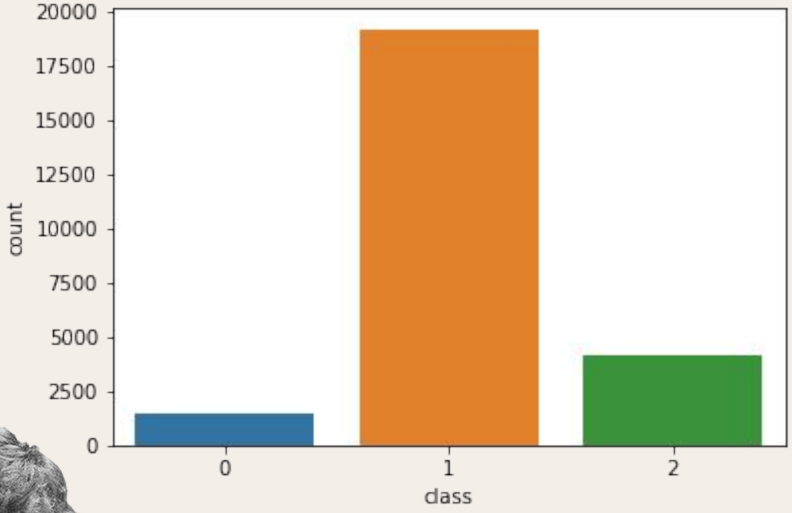
class

Label for majority of CF users

tweet

Text of the tweet

Labels



Hate Speech

"We hate niggers, we hate faggots and we hate spics"

Offensive Language

“RT @HerMoufPiece: Hairy pussy bitch you the type that got herps”

Neither

“Got my vans on.. My pockets chunky”

Framework

01

Pre Processing

02

Text Classification

03

Text Clustering

Pre-Processing

Binary labels

Reduction of label to a **binary variable** → `type` with 1 for `hate_speech` and `offensive_language` and 0 for neither of them

Lowercase

Reduction of characters to **lowercase**

Useless Characters

Dropping of *urls*, *mentions*, *punctuation*, *emojis*, *numbers* and *extra white space*.
Removal of *repeating characters* (more than twice)

"Amp" & NaN

Correction of a **typographical error**: "Amp" → "And"
Removal of 2 rows with NaN values

Tokenization

Tokenization of tweets

Before Pre-Processing

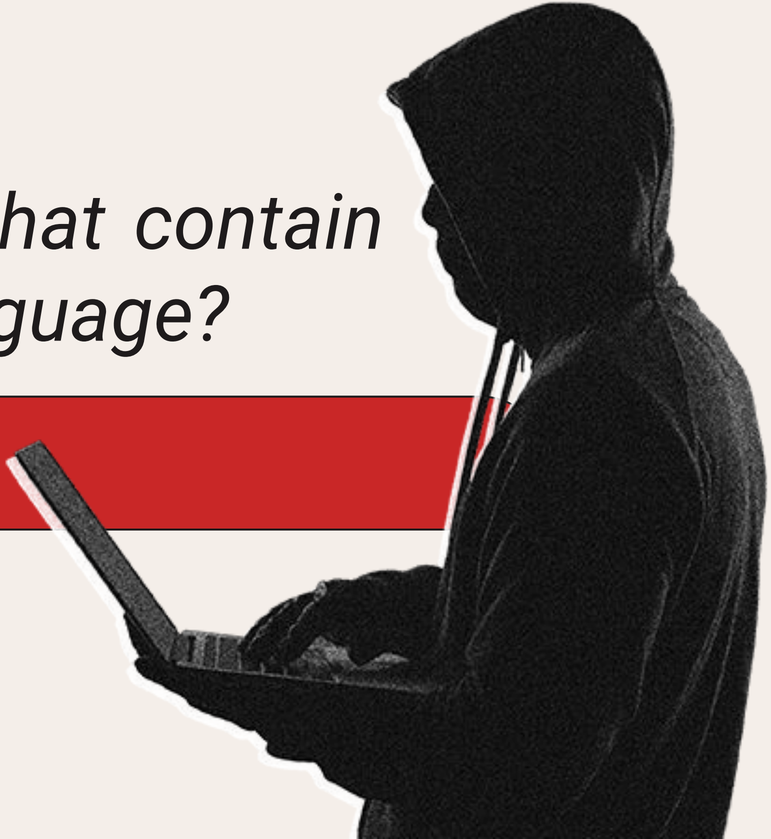
- "We hate niggers, we hate faggots and we hate spics"
- "RT @HerMoufPiece: Hairy pussy bitch you the type that got herps"
- "Got my vans on.. My pockets chunky"

After Pre-Processing

- ['We', 'hate', 'niggers', 'we', 'hate', 'faggots', 'and', 'we', 'hate', 'spics']
- ['Hairy', 'pussy', 'bitch', 'you', 'the', 'type', 'that', 'got', 'herps']
- ['Got', 'my', 'vans', 'on', 'My', 'pockets', 'chunky']

Can we distinguish tweets that contain hate speech or offensive language?

Text Classification



Text Classification

Dataset

The dataset is divided in:

- **70% training set** → 17346 rows
- **30% test set** → 7435 rows

Fixing imbalanced classes in the **training set** with Smote (Synthetic Minority Oversampling Technique) method.

Pre-Processing

Pre-Processing is applied only on the **training set**. The tweets of the **test set** are only removed of the **punctuation**.

Text Representation

TF-IDF

- For using this representation the **STOPWORDS** are removed in the training set
- Using of **lemmatization** on tweets already divided in tokens
- Using of n-grams (**unigram**, **bigram**, **trigram**)

Word2vec

- Using of **unigram**
- **Google pretrained model**

Text Classification with TF-IDF

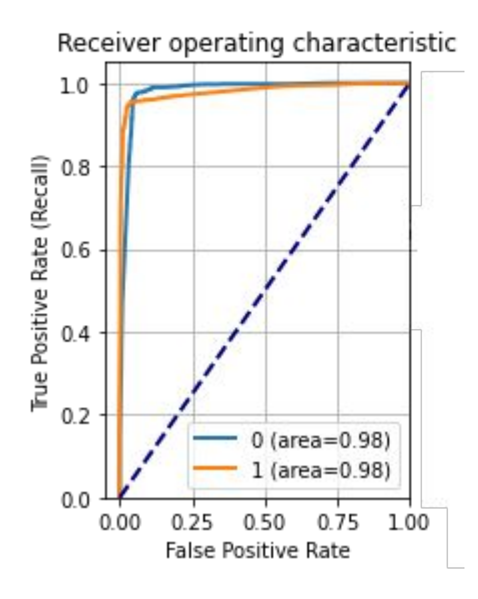
	XGBOOST	Naive Bayes	SVM
Unigram	Accuracy: **0.74** AUC: **0.89**	Accuracy: **0.84** AUC: **0.93**	Accuracy: **0.78** AUC: **0.90**
Bigram	Accuracy: **0.73** AUC: **0.89**	Accuracy: **0.75** AUC: **0.93**	Accuracy: **0.74** AUC: **0.88**
Trigram	Accuracy: **0.73** AUC: **0.89**	Accuracy: **0.75** AUC: **0.92**	Accuracy: **0.73** AUC: **0.88**

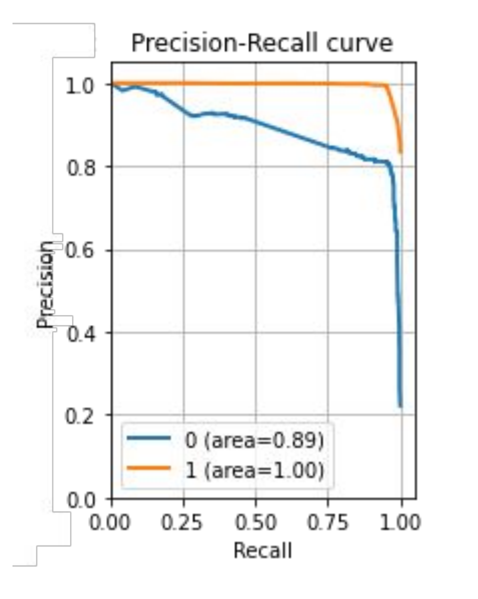
Text Classification with Word2vec

	XGBOOST	Naive Bayes	SVM
Unigram	Accuracy: **0.95** AUC: **0.98**	Accuracy: **0.90** AUC: **0.92**	Accuracy: **0.95** AUC: **0.97**

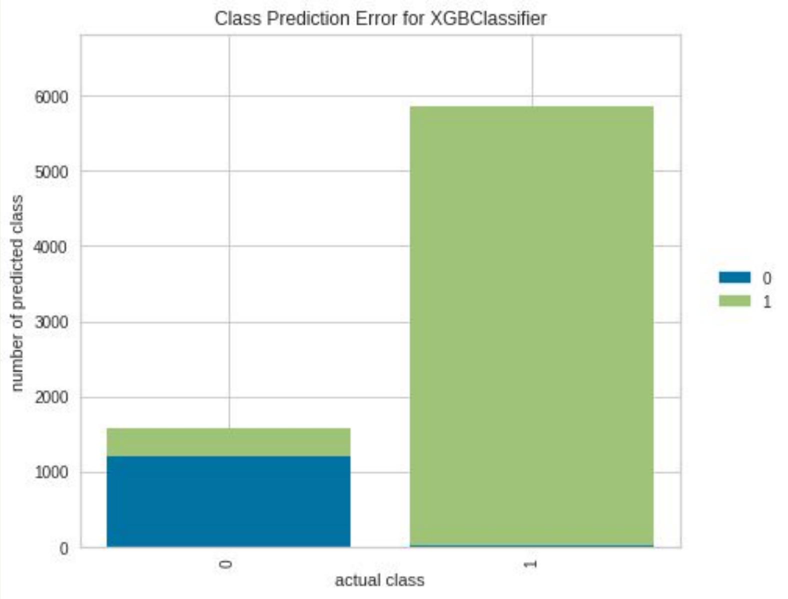
Word2vec - XGBOOST

Results on training set





Results on test set



Is there any division of tweets into clusters?

Text Clustering

Text Clustering

Dataset

A **sample** is extracted from the dataset. This sample respects the **imbalance** between the new two classes of the labels.

Pre-Processing

- Pre-Processing is applied on **all the dataset**
- Tweets are also removed from the **STOPWORDS**
- After tokenization, **lemmatization** is applied

Text Representation

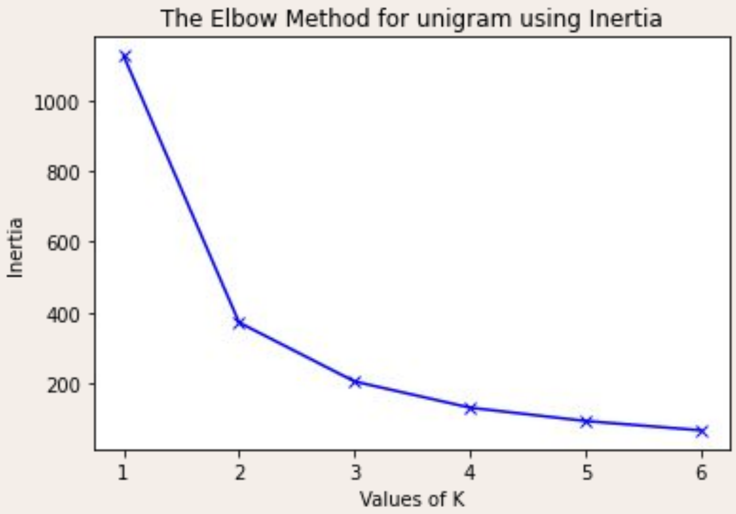
Word2vec

- Using of **CBOW** architecture: `size = 300` and `alpha = 0.03`
- Using of n-grams (**unigram**, **bigram**, **trigram**)

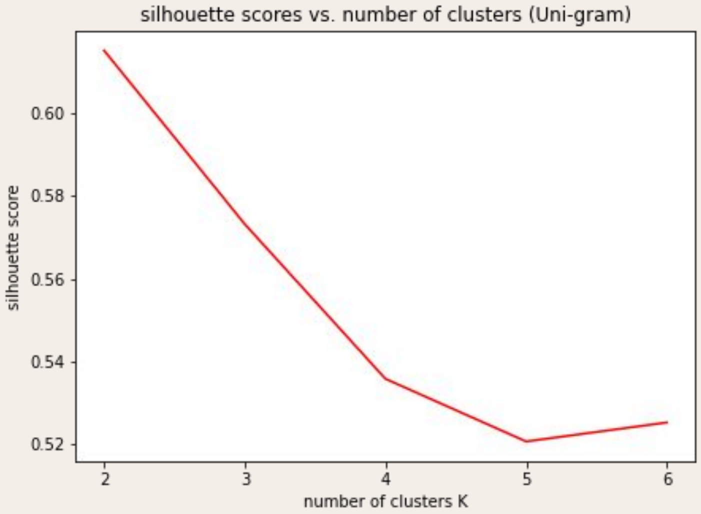
How many clusters?

The Elbow Method and Silhouette Method are computed for each type of n-gram to determine the **optimal number of clusters**. The plots below are about unigrams.

Elbow Method



Silhouette Method

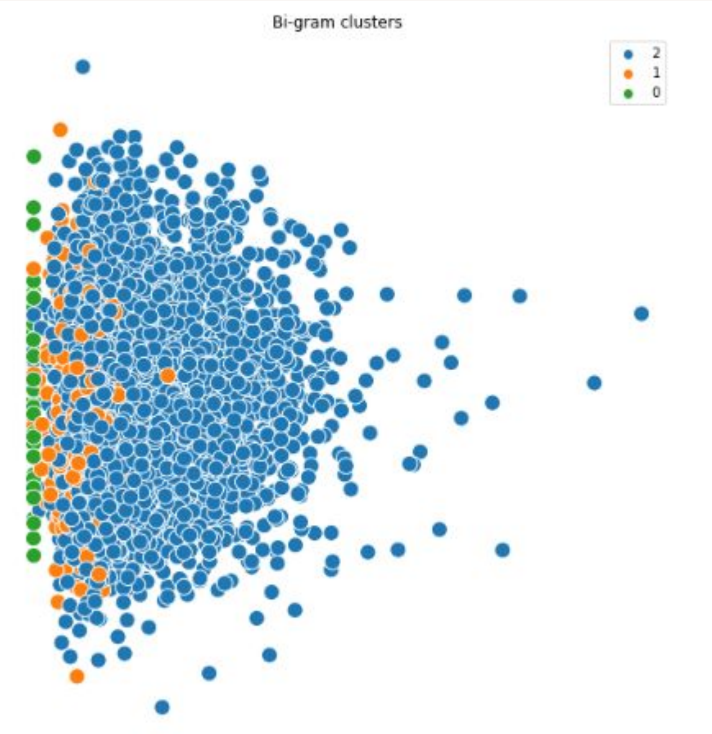


It turns out that **three clusters** are the ideal quantity. The results are analog for bigram and trigram.

Text Clustering with Word2vec

	K-Means	Agglomerative
Unigram	Silhouette index: **0.15** Davies bouldin index: **1.94**	Silhouette index: **-0.24** Davies bouldin index: **0.58**
Bigram	Silhouette index: **0.74** Davies bouldin index: **0.95**	Silhouette index: **-0.59** Davies bouldin index: **0.56**
Trigram	Silhouette index: **0.96** Davies bouldin index: **6.41**	Silhouette index: **-0.73** Davies bouldin index: **0.55**

K-means - Bigram



Top 10 most frequent words for cluster 0

[('trash anyway', 3), ('you pussy', 3), ('whipped cream', 2),('he yank', 2), ('shy people', 2), ('female trash', 2), ('next door', 2), ('oreo milkshake', 2), ('hat ghetto', 2), ('could get', 2)]

Top 10 most frequent words for cluster 1

[('street', 9), ('uncle om', 7), ('hoe lol', 6), ('when i', 5), ('fucking retard', 5), ('fucking pussy', 5), ('jig', 5), ('ring', 5), ('unfollow', 5), ('tryin', 5)]

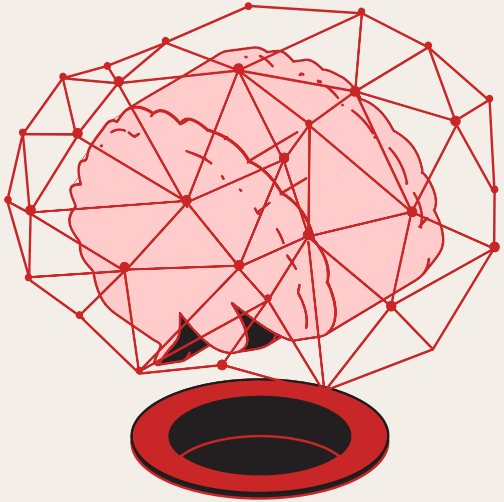
Top 10 most frequent words for cluster 2

[('bitch', 2140), ('i', 1348), ('hoe', 814), ('is', 525), ('like', 517), ('pussy', 401), ('nigga', 350), ('as', 305), ('get', 301), ('but', 282)]

Conclusion

Text Classification

For classification (even if only for unigrams) with the **Word2vec representation** we get a considerable improvement on the results for our dataset



Text Clustering

The available dataset is probably **not suitable** for this type of task since the results obtained in *terms of metrics* seem to be quite satisfactory but *visually* they do not seem to show *particular patterns or features* that allow us to clearly distinguish groups of tweets

Future Developments

Validation Test
Dividing the test set into validation and test set with cross validation

Spelling Correction
Using a function to correct spelling mistakes

Normalization
Using a function to normalize all the vocabulary for all the models

N-grams for Word2Vec classification
Implementing Word2Vec classification also with bigram and trigram

Another dataset
Test the chosen model of classification on another dataset with tweets

Relevant Websites:

- https://medium.com/@dilip.voleti/classification-using-word2vec-b1d79d375381
- https://www.kaggle.com/nlp-model-to-predict-hate-speech#Importing-the-dataset
- https://www.kaggle.com/hate-offensive-language
- https://github.com/Hate-Speech-Detection
- https://medium.com/unsupervised-text-clustering-using-natural-language-processing-nlp
- https://ai.intelligentonlinetools.com/ml/k-means-clustering-example-word2vec/
- https://www.guru99.com/word-embedding-word2vec.html

Relevant Papers:

➢ *Razavi, Amir H., et al. "Offensive language detection using multi-level classification." Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31–June 2, 2010. Proceedings 23. Springer Berlin Heidelberg, 2010.*

➢ *Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the international AAAI conference on web and social media. Vol. 11. No. 1. 2017.*

Thank you for your attention!