Methodology for the challenge

Filipe Afonso

07/10/2018

Data

A train file of 31962 tweets with id, description of the tweet and the label taking two values depending on whether the tweet is considered 0 : not hateful, 1 : hateful

id	label	tweet
1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
3	0	bihday your majesty
4	0	#model i love u take with u all the time in urð±!!! ðððð¦ð¦ð¦

A test file of 17197 tweets. We do not have the "label" column because we have to predict it.

id	tweet
31963	#studiolife #aislife #requires #passion #dedication #willpower to find #newmaterialsâ¦
31964	@user #white #supremacists want everyone to see the new â #birdsâ #movie â and hereâs why

Problematic

Analyzing the train file, predict which tweets in the test file are hateful or not.

Methodology

a. Cleaning of the train and test data files

- string.punctuations, special characters, words with misinterpreted characters, small words (<3characters)

b. Hashtags and “regular” words are separated to allow different weights in the scoring model. Indeed, hashtags are known to be valid words or expressions so that it is good to preserve them. Cleaned train and test files are saved for reuse in the following runs. Data are saved with the following format :

tweet	hate	hashtag
1	0	run
2	0	lyft
2	0	disapointed
2	0	getthanked
4	0	model
5	0	motivation
6	0	allshowandnogo

c. Calculating, for the train file, the number of iterations (support or frequency) of all the words (regular and hashtags) for the ‘hate” labels {0,1}

- pd.crosstab(trainw['word'],trainw['hate']).

d. Deducing empirical conditional probabilities : P(h=1/w)

where h=1 means that hate label=1 and w is :

(1) a “regular” word
(2) or a hashtag
(3) or a sequence of two or three consecutive words (regular or hastag)

item	nb0	nb1	proba
2016election	0	1	100.0
2016in4words	0	12	100.0
ableism	0	4	100.0
abuse	2	2	50.0
administration	0	1	100.0
adultery	0	4	100.0
afghanistan	1	1	50.0

NB : Here I prefer conditional probabilities than Tf*IDF because we particularly want to differentiate tweets with label hate=1

e. Scoring the tweet

We note A the set of “regular” words, B the set of hashtags,

A tweet $t = t_A \cup t_B \in T$ where t_A (resp. t_B) is the set of regular words (resp. hashtags) in the tweet t.

$$score(t) = p_A \times \sum_{a \in t_A} P(h=1|a) + p_B \times \sum_{b \in t_B} P(h=1|b) + p_C \times \sum_{\substack{c_i \in t \\ c_{i+1} \in t}} P(h=1|c_i c_{i+1})$$

with $\sup(a) \geq minSup$ and $\sup(b) \geq minSup$ and $\sup(c) \geq minSup$

$with\ P(h=1|a) \geq minP$ and $P(h=1|b) \geq minP$ and $P(h=1|c_i c_{i+1}) \geq minP$

and where we have 5 given parameters :

- p_A, p_B, p_C are weights according to the importance given to “regular” words, hashtags and sequence of 2 words respectively (optional parameters);
- minSup (minimum support or frequency) and minP (minimum probibility).

The prediction h(t) is deduced with

$$h(t) = 1\ if\ score(t) \geq minScore$$

- minScore : given parameter minmum score

f – Fixing the combination of parameters minSup, minP, minScore

The train file is divided into two files, the first with 2/3 of the data and the second with 1/3 of the data.

The model is calculated, varyng the parameters, with the first file and is tested using the predictions from the second file. The selected list of parameters optimizes the predictions according to the F1 criterion suggested by the challenge :

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

Precision = TP/TP+FP

Recall = TP/TP+FN

g- Application of the model, with the selected combination of parameters, to the test file.