Useful Functions for NLP and Text Analysis

LolloAAA
Lorenzo Ferri

Description

In `Text Analysis` and `NLP` the first and most accurate step is **Preprocessing**. This consists in cleaning the text from words and elements useless for training our model (**Classifier, Neural Network, Clustering...**). Words like `articles, pronouns, prepositions, punctuation, stop-words` are deleted from the original text and preparated for the next step, the training.

While the useless element above are very easy to implement in our algorithm, there are other most difficult to calculate and analyze and **give the ability to a computer to recognize them**.

So I would share with you my work on these particular features and I hope you like it.

Deleting Laughs

The first and simplest algorithm is for detecting and deleting laughs inside a text. Laughs like **ahahahah, ehehehe, ihihihi** can be useful for text analysis, so it's important detecting them. My algorithm control the length of a laugh and if this is formed by some letters (like vowels) and set an offset to, for avoiding typing keyboard error that can make laugh unrecognizable.

`I'm joking ahahhahahaha`$\Longrightarrow$ `I'm joking`

Detecting hidden Bad Words

This second is for recognize **hidden bad words** in the text. For hidden bad words I mean bad words written with special symbols in the middle for make

them unrecognizable from a computer or a security system. E.g. the word "**asshole**" can be written as "**as....e**".
I use a list of bad words and not a large dataset, because only few and specific words are used for this scope. But if you want expand the possibilities, you can use it.

```
You are an as..--.-xxxle
⇓
You are an asshole
```

Splitting Hashtag

The last and most interessant is for splitting an hasthag contained in a text. Is very difficult for a computer recognize the subwords. So for this algorithm I used a very large vocabulary (list) of all words in **.csv** format. If you want use this, I hint you to find a large dataset of your language and import this in the script. It choose the words by their length and orderly presence in the dataset, so if it find an ambiguity in the hastag, It could split the hastag in another set that you didn't expect it. But my algorithm does its work and this is a future work to be completed and improved.

```
#nelMezzoDelCamminoDiNostraVita
⇓
nel mezzo del cammino nostra vita
```