Universidad de Castilla-La Mancha

Escuela Superior de Informática

Natural Language Processing

GitHub

Authors:

Josue Carlos Zenteno Yave
Sergio Silvestre Pavón
Javier Álvarez Páramo
Sergio Martín-Delgado Gutiérrez

1 Introduction

In real life there are so many websites where you can write reviews about cars, toys or any other things, but, there is a problem with this and it is that whenever you write a comment you do not do it in a proper way. So trying if you try to analyze these data you can find many problems.

For this work our team counts with a data-set with hundreds of reviews collected from a series of web pages that talk about both, cameras and autos.

So the final purpose of this work is trying to analyze the data-set using NLP (Natural Language Processing) algorithms in order to be able of training a model that will be capable to classify a review saying if it is a camera review or an auto review.

2 Preprocessing

At this section all the considerations that have been taken to normalize and clean the data of the data-set are going to be explained. This explanations are also a part of the own colab file so, if you want to read them there, you can do it easily.

2.1 Removing useless data

First things first, as into the data-set there are real reviews written by people, it is easy to find some grammatical errors, missing words, emojis, special characters...

Therefore, in order to turn this data into something useful and make it a valuable thing we have to clean the data. So, at this point we are going to remove all the punctuation marks from the different reviews.

We have considered all the English punctuation marks. So, in order to remove them, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.2 Removing all capital letters

Once we have deleted all the punctuation marks we can proceed by removing all the capital letters that exists in the data of the data-set. As we are using Python3 we can easily turn all the text from capital letter to lower case by using a simple function.

So, in order to remove them from all the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.3 Lemmatizing all terms

As we have successfully removed all the punctuation marks and capital letter we can now lemmatize all the words that are part of the different sentences that we have into the data-set. To do that, first we have to explain what do we mean by saying "lemmatize". Lemmatizing is not something strange, in fact it is as easy as you can see in the following example: Ex: Working – Work, Tried – Try.

In conclusion, lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning. So, we have to do this because we do not have any profit by using the conjugated words so by removing them we simplify the data.

In order to remove them from all the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.4 Removing contractions

Another important thing, is trying to remove all the contractions that appears in the text. This is because of the same reason than the previous step, because we do not have any kind of profit by having contractions in our data.

So, in order to remove them from all the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.5 Removing repeated words

Now, another important step is to remove the repeated words in the text, usually English speakers emphasize what they say by repeating words but now we do not need that in our text.

So, in order to remove them from all the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.6 Removing or replacing emoticons

We think that, at this point, the purpose of removing emoticons or "emojis" is obvious. It is something irrelevant in our task and we have to remove them in the same way we have removed the previous kind of irrelevant information.

So, in order to remove them from all the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

2.7 Correcting wrong words

As we said before, people doesn't care about having a perfect grammar when they write a review. People usually writes them in a fast way with lots of misspellings. Now in contrast to the rest of the sections having our text properly written is very important thinking about the following steps (model ones)

So, in order to correct the grammatical errors in the reviews, an algorithm has been developed and you can find it in the corresponding section into the colab file.

3 Vectorization

In this section we are going to explain in detail how do we have vectorized all the data that is contained in the data-set, obviously this data has already been cleaned and normalized in order to be able to work with it in an easier way.

Something that we want to remark is that we have applied different vectorizers in order to achieve the best results (as it will be shown later).

3.1 TFIDF

TFIDF (Term Frequency - Inverse Document Frequency) measures the relevance that a word has in a document and this relevance can be calculated by applying the following mathematical expression.

$$TF_{(i,j)} = \frac{n_{(i,j)}}{\sum n_{(i,j\ )}}$$

Where $n_{(i,j)}$ is the number of times that a word appears in a document and $\sum n_{(i,j)}$ is the number of words in the document.

$$IDF = 1 + \log(N/dN)$$

Where 1 could or could not be in the expression, if it appears it is called smoothing. N = Number of documents y dN = Total number of documents where a word appears.

So $TFIDF$ is the product of multiplying TF by IDF, resulting on the original mathematical expression.

An algorithm has been developed and you can find it in the corresponding section into the colab file.

3.2 TFIDF + N-grams

Once we did the TFIDF, we are going to go a step further: the inclusion of N-grams. N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document. So the TFIDF will be done for a established range of N-grams, having terms of different size.

An algorithm has been developed and you can find it in the corresponding section into the colab file.

3.3 TFIDF + N-grams + POS tagging

Having the relevance that a word or a sequence of words have, the next step is to assign the grammatical category. For that reason, we use the POS tagging technique. More precisely, Pos tagging is the process of marking up a word in a text as corresponding to a particular part of speech, based on both its definition and its context.

An algorithm has been developed and you can find it in the corresponding section into the colab file.

4 Feature selection

In this section we are going to explain the process for the selection of features in the data-set obtained previously. Before that, the data has already been cleaned, normalized and vectorized in order to be able to work with it in an easier way.

The process of selecting the best features will be done using the selecKBest and removing 70% of the features we have in our data-set. So for the selecKbest algorithm we will indicate that the 30% are the selected.

So, in order to select the best features, an algorithm has been developed and you can find it in the corresponding section into the colab file.

5 Classification algorithm

In this section we are going to explain the approach for the classification algorithm. Before that, the data has already been cleaned, normalized, vectorized and with the best features selected in order to be able to work with it in an easier way.

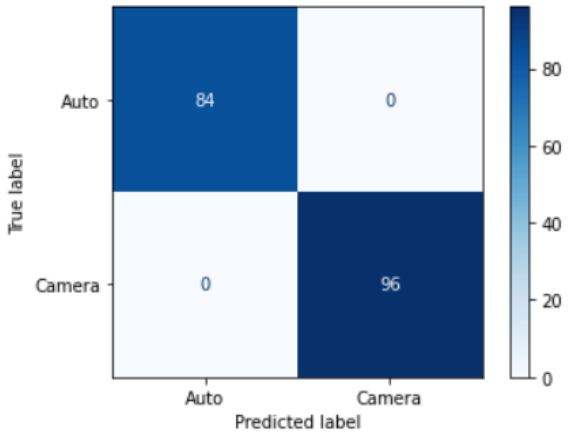
The classification of the opinions will be done differentiating between auto (0) and camera (1). To accomplish that, 70% of the data-set obtained after selecting the best features will be used for training and 30% for testing. Finally, for the classification a SVM model will be applied.

So, in order to do the classification, an algorithm has been developed and you can find it in the corresponding section into the colab file.

6 Experiments and results

In this section we are going to evaluate some results that we considered to be meaningful. Before that, the data must complete all the process in order to be able to work with it in an easier way.

First of all, we are going to study the confusion that we can observe in the image below. There we can deduce that everything is correct, all the elements in their corresponding place, nothing about false positive or false negative.



After that, we appreciate some measures. This measures are:

- Precision: measure the quality of the model in classification tasks.
- Recall: inform about the quantity that the model is able to identify.
- F-measure: is used to combine the precision and recall measurements in only one value. With this value it is assume that the precision and the recall have the same importance.

In all this metrics we obtained the best possible value. So, we can conclude that our model can give great performance classifying and identifying the elements.

To end, we also consider to evaluate the different errors and as it happened with the previous metrics we obtained the best possible results.

All this calculations has been done in the colab and you can find it in the corresponding section into the colab file.