NLP PROJECT

Sentiment Analysis

Luis Valdivielso Capell

Luis.valdivielso.capell@alumnos.upm.es

INDEX

1. INTRODUCTION 3
2. PROBLEM 3
3. METHODOLOGY 3
4. RESULTS AND DISCUSSION 4
5. CONCLUSION 7
References 8

1. INTRODUCTION

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers. Natural language processing strives to build machines that understand and respond to text or voice data and respond with text or speech of their own in much the same way humans do. (1)

There are many types of projects based on NLP, for example: Chatbots, Grammar Autocorrector, Spam Classification, Sentiment Analysis... In this paper we will deal with the latter (2)

2. PROBLEM

Sentiment Analysis is a process of extracting opinions that have different scores like positive, negative or neutral. Based on sentiment analysis, you can find out the nature of opinion or sentences in text. Sentiment Analysis is a type of classification where the data is classified into different classes like positive or negative or happy, sad, angry, etc. This type of analysis can be useful to find out what kind of ratings a certain product has, we can discover which words are the most repeated when rating that certain product, if the comments are positive or negative. (3)

For this work we have 2000 reviews put on the official website of amazon, of the book Ikigai: The Japanese Secret to a Long and Happy Life. Our aim is to try to extract as much relevant information as we can from analysing these data.

3. METHODOLOGY

The next question we ask ourselves, once we have collected the data we are going to use and once we have described the problem, is how we are going to analyse the data.

First of all, we are going to mention which tool we are going to use, for this kind of analysis we usually use environments that handle Python and environments that handle R, in our case we will use Rstudio that supports R.

In order to carry out the analysis we will proceed as follows. First of all, we construct a corpus, a corpus is a collection of text document over which we would apply text mining or natural language processing routines to derive inferences, then we will do a data cleansing, which consists of eliminating certain characters that are not necessary and words that do not provide us with relevant information. In addition, the structure of the text is changed to make it easier to handle by certain algorithms (Normalisation, all text in lower case, ...).

We will use the following packages to manipulate and extract information from the data: (3)

- **tm** for text mining operations like removing numbers, special characters, punctuations and stop words (Stop words in any language are the most commonly occurring words that have very little value for NLP and should be filtered out).
- **wordcloud** for generating the word cloud plot.
- **syuzhet** for sentiment scores and emotion classification
- **ggplot2** for plotting graphs
- **e1071** for build and train SVM model

4. RESULTS AND DISCUSSION

In this section we will carry out all the phases mentioned in the methodology using Rstudio, as mentioned previously.

```
> reviews<-read.csv(file.choose(),header=T)
> str(reviews)
'data.frame':   2000 obs. of  8 variables:

$ id          : chr
$ profileName: chr
$ text        : chr
c"| __truncated__ "
do"| __truncated__
 u"| __truncated__
$ date        : chr
en EspaÃ±a el 5 de
$ title       : chr
$ rating      : int
$ images      : chr
$ helpful     : int
```

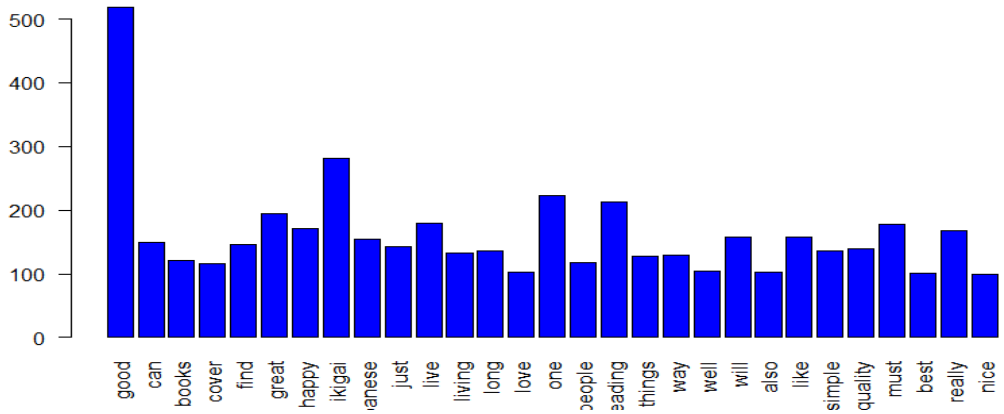
Importing the csv, we can see that we have 2000 observations and 8 variables, in our case we are only interested in the variable "text" which is the one that contains the text of each of the reviews left by the Amazon users.

The next step is to create the corpus and perform a cleaning of the data, eliminating capital letters, punctuation marks, numbers, unnecessary spaces... as well as some other words that we know are going to be repeated a lot and are not going to provide relevant information to analyse if a review is positive or negative. (3)

```
#Set corpus
corpus<-iconv(reviews$text[-1])
corpus<-Corpus(VectorSource(corpus))
inspect(corpus[1:5])
#cleaning corpus
corpus<-tm_map(corpus,tolower)
corpus<-tm_map(corpus,removePunctuation)
corpus<-tm_map(corpus,removeNumbers)
corpus<-tm_map(corpus,removeWords, stopwords("english"))
corpus<-tm_map(corpus,removeWords, c("book","read","life"))
corpus<-tm_map(corpus,stripwhitespace)
```

(in the first line we use reviews$text[-1] to not use the first review, as we have been able to check that it was in Spanish and it is not necessary to analyse it).

We can use a barplot to check which words are the most repeated (all the words that appear in the barplot have been repeated 100 times or more):



As we can see, the word that is repeated most often is undoubtedly "good", which reflects the fact that the book is generally rated positively.

Another way of visualising which words have more weight or appear more frequently is to use a word cloud:

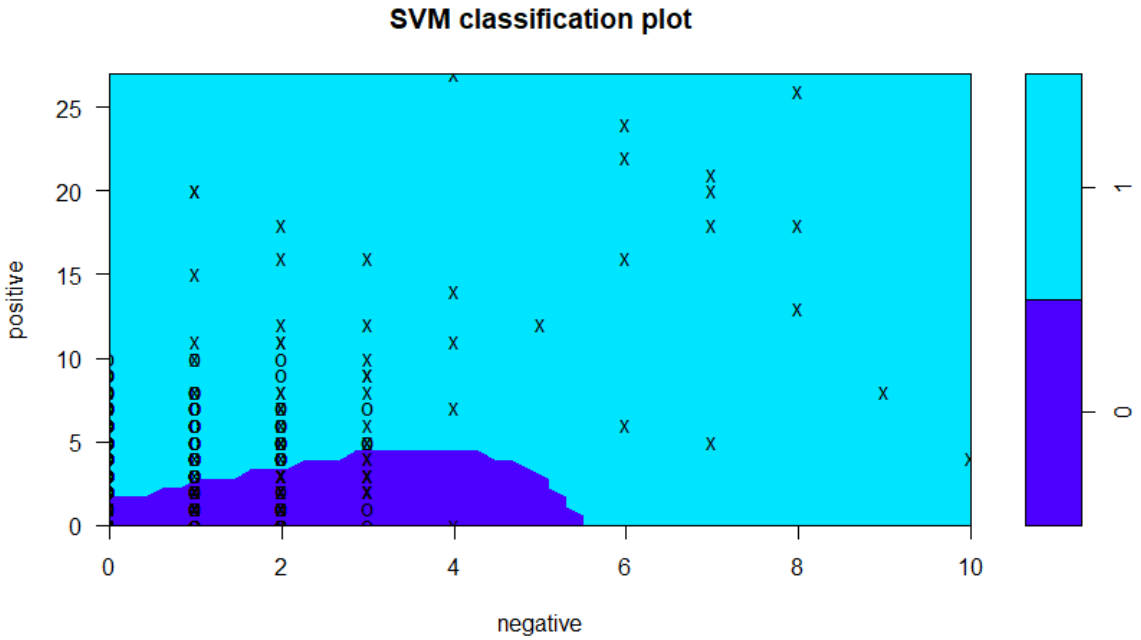
We can see again that "good" is the word that is most frequently repeated by observing its size in the word cloud, in neither of the two graphs can we observe words with a negative meaning, so there should not be a very high frequency of negative ratings. In general terms, the ratings are good, but it is quite likely that there are negative or mixed ratings that we are not able to see in the graphs because only the most frequent words are represented. We will use the "syuzhet" package to obtain the relative scores for each of the reviews.

```
> s[1:10,]
   anger anticipation disgust fear joy sadness surprise trust negative positive score
1      0            1       0    0   3       0        2     2        0        5     5
2      0            0       0    0   0       0        0     1        0        0     0
3      0            0       0    0   0       0        0     0        0        0     0
4      0            1       0    0   1       0        1     1        1        1     0
5      2           10       1    4   8       9        2    12        7       20    13
6      1            7       1    3   9       3        3    11        6       24    18
7      0            0       1    0   0       0        0     0        1        0    -1
8      3           11       1    2  11       2        5    13        8       18    10
9      2            5       1    0   4       2        3     6        4       11     7
10     3           14       1    2   9       3        1    13        6       22    16
```

We can see in the image above, the scores of the first 10 reviews, the last column represents the score, which is the difference between positive and negative emotions, the higher the number, the more positive that review is, of the first 10 we can see that the 6th rating is the most positive and the 7th is the most negative.

Once we have collected the scores of each of the observations, we will try to train a model that helps to distinguish them graphically between positive and negative, for this we will use the SVM, A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model training data sets labelled for each category, they are able to categorise a new text. Compared to other more recent algorithms, such as neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands).

This makes the algorithm very suitable for text classification problems, where it is common to have access to a dataset of at most a couple of thousand labelled samples.



We can distinguish between positive and negative reviews by looking at the colour of the graph, in the area coloured light blue are the positive reviews, i.e. they have a score (number of positive words - number of negative words) positive >0 and in the other area are the negative reviews which have a score <=0. We can see how most reviews are positive and many of them have a large number of positive words (high Y-axis), the few negative reviews have few negative words (X-axis).

We can see the overall score obtained by the model by looking at its confusion matrix and its main reference measures:

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  658    1
         1    0 1340

               Accuracy : 0.9995
                 95% CI : (0.9972, 1)
    No Information Rate : 0.6708
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9989

 Mcnemar's Test P-Value : 1

            Sensitivity : 1.0000
            Specificity : 0.9993
         Pos Pred Value : 0.9985
         Neg Pred Value : 1.0000
             Prevalence : 0.3292
         Detection Rate : 0.3292
   Detection Prevalence : 0.3297
      Balanced Accuracy : 0.9996

       'Positive' Class : 0
```

T

We can see that it scores very well on all measures.

5. CONCLUSION

Once we have reached this point, we can review the main conclusions that have been obtained from this work. First of all we have compiled the text we were going to analyse from a series of reviews, we have analysed and cleaned the data to be able to handle it better. The score that book has on Amazon is 4 and a half stars out of 5, we have been able to check something similar by obtaining the frequency of each word among the 2000 reviews we have analysed and using the algorithm provided by the syuzhet library to obtain the score of each review according to the number of positive and negative words that each review has. Each of these procedures showed similar results, a large number of positive reviews.

Finally, we have trained an SVM model from the e1071 library and the data set collected with each of the scores obtained as mentioned above, obtaining very good results for the model, capable of predicting almost entirely a negative review from a positive review.

References

1. **https://machinelearningmastery.com/natural-language-processing**

2. https://www.projectpro.io/article/nlp-projects-ideas-/452.

3. https://www.r-bloggers.com/2021/05/sentiment-analysis-in-r-3