IMDB Reviews

Text Mining and Search

Agazzi Ruben 844736, Cominetti Fabrizio 882737

Abstract
In this project, user reviews from the IMDB platform were analyzed through the use of text mining techniques. After carrying out an initial phase of text processing and text representation, the project continued with the classification of the reviews, through some text classification techniques - such as Support Vector Machines (SVM), Multilayer Perceptron (MLP), and Logistic Regression. Next, a text clustering phase was carried out through the use of two algorithms: DBSCAN and k-means.

Keywords
Text Mining — Text Classification — Text Clustering

University: University of Milan-Bicocca

Contents

1. Introduction

The project aims to analyze the dataset ”IMDB reviews” through text mining techniques, specifically through *Text Classification* and *Text Clustering*. The dataset contains a total amount of 50000 user-released reviews on the IMDB platform, divided in half between training and testing. The dataset is ideal for performing a binary sentiment classification task, the first objective of our analysis. Next, we decided to exploit text clustering techniques with the goal of identifying different clusters within the text.
Text Classification is the activity of predicting which data items belongs to a predefined finite set of classes. There are many types of classification, in our case it is *Binary Classification*, where each item belongs to exactly one class in a set of two (positive or negative).
In addition, text classification may be performed according to several dimensions (’axes’) orthogonal to each other. For example, by topic (the most frequent case), by sentiment - our case -, by language, by type, by author, by native language, by gender, and more.
Text clustering, on the other hand, is the task of grouping a set of unlabeled texts in such a way that texts in the same cluster are more similar to each other than to those in other clusters.

2. Data

As stated before, the dataset used contains a total of 50000 user reviews on the IMDB platform, a platform that describes itself in the following manner :”IMDb is the world’s most popular and authoritative source for movie, TV and celebrity content. Find ratings and reviews for the newest movie and TV shows” [**IMDB**].
The dataset is also defined as a ”Large Movie Review Dataset”. From an initial exploration of the data, we can observe that the dataset does not provide information about the date and reference film of the review, or any other indication, but contains only the text of the review and the extracted sentiment - positive or negative.
The data, initially divided into training and testing, but also between positive and negative sentiment, were merged, so that there would be a single dataset for the 25000 reviews to be used in the training phase and the 25000 reviews to be used in the testing phase.
Finally, the dataset contains precisely 12500 reviews labeled as positive and as many labeled as negative, both training and testing.

3. Text Processing

Having obtained the starting dataset, a series of *Text Processing* operations were performed:

- *Remove Numbers*, all numbers within the text have been removed;
- *Remove StopWords*, all words in the stopwords list have been removed;
- *Remove Punctuation*, all punctuation has been removed;
- *Remove Extra Space*, all extra spaces within the text have been removed;
- *Tokenization*, the process of breaking down a text into units called tokens;
- *Lower Case*, all words were converted to lower case;
- *Lemmatization*, the process of grouping together the inflected forms of a word.

Once the corpus of texts had been properly processed, we moved on to the next stage of text representation.

4. Text Representation

Text Representation is the process to represent text with graphical methods. Considering the purposes of the project, the reviews were represented in structured form according to two methods: *Bag of Words* and *Tf-Idf*.
The Bag of Words representation identifies each document by a vector in which contains the number of occurrences of each word. This model doesn't consider grammar and order of words.
In the Tf-Idf representation, the number of occurrences of each word is weighted against the inverse of the word's presence in the corpus.
The weights, called *Tf-Idf* weights, are the product of the two indices *Tf* and *Idf*:

$$w_{t,d} = \frac{tf_{t,d}}{max(tf_{t_i,d})} \times log(\frac{N}{df_t})$$

Where the Term Frequency $tf_{t,d}$ represent the frequency of the term t in the document d, divided by the frequency of the most occurring word in the document to prevent bias towards longer documents; and the the Inverse Document Frequency idf_t represents the inverse of the informativeness of the document for a term t.
The two representations were implemented through two features of the sklearn package in python: CountVectorizer for Bag of Words, TfidfVectorizer for Tf-Idf. The range of n-grams chosen for both was 1-2, that is, Uni-grams and Bi-grams.
Below we can observe some basic statistics for the train and the test set. Precisely, the number of words in the corpus, and the average review length.

	Train	Test
Number of words	24902	24798
Average review length	685.01	668.02

Table 1. Statistics for train and test data

5. Text Classification

...

5.1 SVM

...

5.2 Multilayer Perceptron

...

5.3 Logistic Regression

...

5.4 Evaluation

...

6. Text Clustering

...

6.1 DBSCAN

...

6.2 K-means

...

6.3 Evaluation

...

7. Summary

...