NLP techniques applied to Amazon reviews

Boller Mattia - 873358

Text Mining and Search
Master course in Data Science, University of Milano Bicocca

Abstract: In this project, different NLP techniques were used to process a set of Amazon reviews, written by buyers. Documents have been preprocessed by applying urls, emojis, numbers, punctuation and stopwords removal. Every word have been then POS tagged and lemmatized. The resulting texts have been represented with different methods, in particular with Tf-Idf representation, Word2Vec representation and Doc2Vec representation. The vector representations of the documents have been used to compute:

- Text classification: different machine learning methods were used (Naive-bayes, Support Vector Machine, Logistic Regression) and the results were compared using ROC curve and AUC score. Best results were given by the Tf-Idf representation and by using SVM and Logistic Regression algorithms, with and AUC score of 0.90 and an accuracy of 0.83.
- Text clustering: clusters built with Kmeans algorithm, optimal number of cluster found through elbow method, extraction of clusters topics through clusters centroids coordinates and visualizations. It was possible to identify 5 different clusters described by the following topics: Videogames, Movies, Products, Music, Books.

Keywords: NLP, text representation, text classification, text clustering.

1 Introduction

Amazon.com is an international e-commerce company that offers online retail, computing services, consumer electronics, digital content, and other local services such as daily deals and groceries. Amazon is the leading e-retailer in the United States, with net sales amounting to close to 386 billion U.S. dollars in 2020 [1]. The company ships approximately 1.6 million products per day and buyers can leave a textual review for those products, leading to the generation of a very large amount of text.

In this project the objective was to apply to a certain amount of these documents generated by the customers some natural language processing techniques, in particular sentiment analysis techniques and text clustering techniques. Before performing these tasks, different types of text representation were applied to the reviews to understand which could lead to the best results the case under consideration.

2 Data

In the following paragraphs are described the raw data used in the project and all the preprocessing operations applied to it.

2.1 Raw data

The original Amazon reviews dataset consists of 35 million reviews from amazon. The data span a period of 18 years, up to March 2013. The dataset used in this project is the one created by Xiang Zhang by randomly taking 600,000 training samples and 130,000 testing samples for each review score from 1 to 5 from the dataset described above and used for the the paper ”Character-level Convolutional Networks for Text Classification” [2].

Figure 1: Wordclouds for positive and negative reviews

The dataset is in csv format and contains the text of the reviews, the titles and the rating score between 1 and 5. For the project only 300,000 reviews were sampled to keep a reasonable code execution time.

2.2 Data preprocessing

The first operation applied to the dataset, was creating a new column "sentiment", filled with 1 if the review score was lower than 4, filled with 0 otherwise. This new feature became the objective of the sentiment analysis. After this operation, there was 180,003 negative sentiment reviews and 119,997 positive sentiment reviews, so the set was slightly unbalanced. In figure 1 are presented the wordclouds for positive and negative reviews.

The next preprocessing operations aimed to clean the texts of the reviews and reduce the vocabulary by reducing the words to inflectional forms. From the documents were removed urls, emojis, numbers, punctuation and common english stopwords. Some stopwords were not removed because of possible useful information during sentiment analysis, for example "not" and "but". After this first cleaning, lemmatization was applied to every word, that is the process of determining the lemma of a word based on its intended meaning. To effectively apply lemmatization, it is essential to understand the true meaning of every word based on the context, so POS tagging (Part of speach tagging) was applied to the reviews and a tag (adjective, noun, verb, adverb) was assigned to every word. Lemmatization was preferred to stemming given the importance of the meaning of the words in both sentiment analysis and text clustering, for better understand the topics of the clusters.

3 Text representation

In the following paragraphs are presented the different text representation techniques used to transform text documents into numerical vectors. Three different methods were tested for research purposes. The dataset was split into train and test sets and representations models were built on the training set to simulate a real case were new data need to be processed.

3.1 Tf-Idf

The first text representation tested was the Bag-of-Words with weighted numeric representation of the words given by Tf-Idf. Tf-Idf (term frequency–inverse document frequency) is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [3]. Tf-Idf factor is computed in the following way:

$$w_{t,d} = \frac{tf_{t,d}}{max_{ti}(tf_{ti,d})} \cdot \log_{10}(\frac{N}{df_t}) \qquad (1)$$

where $tf_{t,d}$ is the frequency of the term t in document d, N is the total number of documents and df_t is the number of documents that contain the term t.

For computing the Tf-Idf matrix, unigrams and bigrams were considered, for a maximum of 10,000 features and with a minimum document frequency of 5.

3.2 Word2Vec

The second text representation tested was based on the use of Word2Vec model. The method consist on using a neural network to learn a representation of the words. There are 2 architecture that can be used to train the model:

- Skip gram: predict the surrounding words (context) based on the current word.
- CBOW (Continuos bag of words): predict the centered words given the context words.

In both cases, the embedded representation of the word is given by the hidden layer of the neural network. In this project the CBOW model was chosen, with a vector size of 300.

After the model was built, it was used to compute the representation for every word in every review and the final representation of the document was computed as the average vector of the vectors of the words of which it was composed.

3.3 Doc2Vec

The third text representation tested was based on the use of Doc2Vec model. Doc2Vec can be considered an extension of the Word2Vec model described in the paragraph 3.2 and it is used to compute vector representations of documents instead of words. The algorithm represents each document by a dense vector which is trained to predict words in the document [4].

4 Text classification

The different text representation techniques presented in the section 3 were used to obtain vectors representing all the documents. These data were used as input for training and testing different models for text classification. The problem has been approached as a binary classification task, with the 0 class representing negative sentiment and the 1 class representing positive sentiment. The models that have been tested are:

- Naive Bayes
- Support Vector Machine
- Logistic Regression

Every classifier has been trained with 70% of the original dataset and tested with the remaining 30%. The accuracy score obtained by every model with every text representation can be seen from the table 1.

	Tf-Idf	**W2V**	**D2V**
Naive Bayes	0.80	0.61	0.60
Support Vector Machine	0.83	0.79	0.73
Logistic Regression	0.83	0.79	0.73

Table 1: Accuracy values for every classifier with every text representation

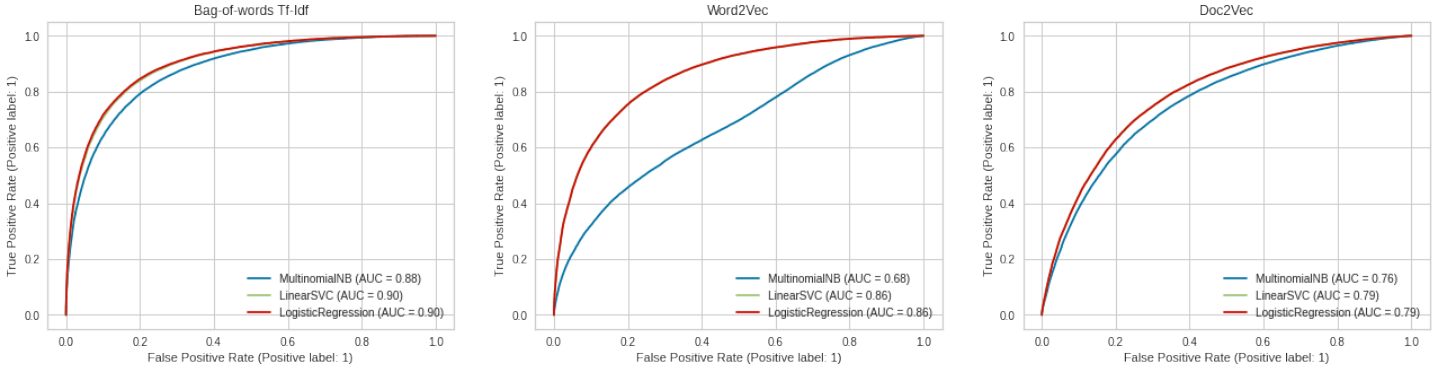

Figure 2: ROC curve for every classifier

Since the dataset is slightly unbalanced, the accuracy is not the perfect metric to compare the models, so AUC score was computed too. The figure 2 present the ROC curve and AUC score for every classifier, as it can be seen, Support Vector Machine and Logistic Regression trained on the Tf-Idf text representation obtained the highest performances with an AUC score of 0.90.

5 Text clustering

The objective of text clustering was trying to divide the reviews into their respective topic, in a way that each cluster contained reviews of a specific type of product.

For this task, the Bag-of-Words Tf-Idf was used as text representation, with vectors of size 10,000. For hardware limitations, dimensionality reduction was applied using truncated SVD (singular value decomposition). This method performs linear dimensionality reduction by means of truncated singular value decomposition. Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with sparse matrices efficiently, matrices like the ones obtained by Tf-Idf representation [5]. The resulting vectors after the use of truncated SVD had a size of 100.

K-means was the algorithm chosen for the clustering task.

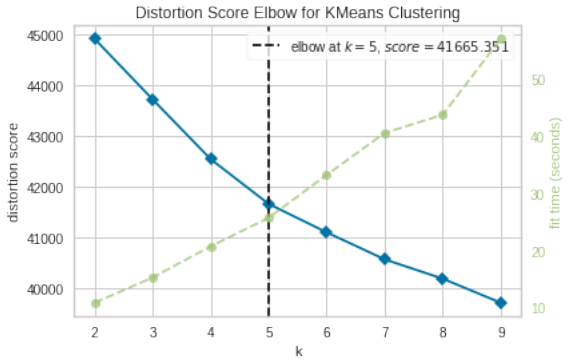

Figure 3: Optimal number of clusters with elbow method

The optimal number of clusters was selected by using the elbow method, using the distortion score as metric to be plotted to find the elbow. The plot of the elbow method can be seen in figure 3.

The silhouette score of the cluster division obtained with K-means algorithm is 0.06.

To visualize the clusters obtained, truncated SVD was applied to the data to get just 3 dimensions, to be able to plot the documents in a 3 dimensional space. The scatter plot can be seen in figure 4. As it can be seen, 4 clusters are clearly divided, the points belonging to the cluster 0 instead, in this visual representation, are scattered among the other clusters.

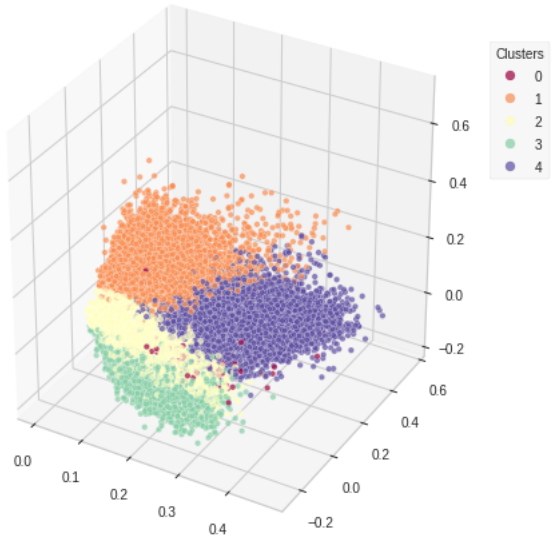

Figure 4: Representation on a 3-dimensional space of the clusters

To understand the content of the different clusters, centroids coordinates were extracted and ordered by magnitude. Coordinates represents the Tf-Idf value of words, so a coordinate with high value means high importance of the respective word. By using this method, the following topics were extracted from every cluster:

- Cluster 0: game, play, be, fun, but, get, graphic, like, not, good.
- Cluster 1: movie, be, watch, but, see, film, not, good, like, great.

- Cluster 2: be, not, but, use, get, work, product, good, would, buy.
- Cluster 3: album, cd, song, be, music, like, but, listen, sound, good.
- Cluster 4: book, read, be, not, but, story, write, good, like, character.

Clusters 0, 1, 3 and 4 contain reviews about clearly defined and different products, while the cluster 2 seems to contain more general reviews.

6 Conclusions

Different text representation techniques were tested for the task of sentiment analysis on Amazon reviews. In the case of this project, the Tf-Idf representation obtained the best results with Logistic Regression and Support Vector Machine models. Different parameters for Word2Vec and Doc2Vec can be tried in the future to better understand if the Bag-of-Words representation is really the best embedding for the dataset considered in this work.

Text clustering gave interesting results, since the clusters obtained, apart from one, were described by clearly defined topics (videogames, movies, music, books). Only one text representation was used to compute clusters (BOW), in the future others embedding methods can be used as input and others clustering algorithms can be tested.

References

[1] "Amazon - statistics & facts." [Online]. Available: https://www.statista.com/topics/846/amazon/dossierKeyfigures

[2] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[3] A. Rajaraman and J. D. Ullman, *Data Mining*. Cambridge University Press, 2011, p. 1–17.

[4] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[5] "Truncatedsvd." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html