Università degli studi di
Milano-Bicocca

Text Mining and Search

Final Project

Exploration of drugs's reviews

Authors:
Tommaso Redaelli - 830442 - t.redaelli7@campus.unimib.it
Riccardo Merlo - 829805 - r.merlo2@campus.unimib.it

February 13, 2022

Abstract

This project is carried out using the data available on UCI repository [1]. The proposed tasks include the classification of the review rating and a clustering analysis aimed at discriminating groups of drugs. The realization of these begins with a massive preprocessing phase in which several transformations of the text of the reviews are obtained. TF-IDF and Word2Vec are used as document representation techniques. The best results in the classification task are achieved through the use of a neural network, while in the clustering analysis the Gaussian Mixture algorithm is exploited, whose outputs are then explored to allow an interpretation.

1 Introduction

Drugs.com is the largest, most widely visited, independent medicine information website available on the Internet [2]. His dataset is powered by several independent leading medical-information suppliers, including; American Society of Health-System Pharmacists, Cerner Multum and IBM Watson Micromedex. Individual drug (or drug-class) content compiled by these sources is peer reviewed and delivered by Drugs.com. The project goal is to perform two types of analysis: a classification task which tries to predict the drug's rating based on the user's review and a clustering analysis based on each condition in order to better understand how drugs are described by their users.

2 Dataset

The dataset provides patient reviews on specific drugs along with related conditions and a 10 star patient rating reflecting overall patient satisfaction.

It is composed of 6 attributes:

- **drugName** (categorical): name of drug
- **condition** (categorical): name of condition
- **review** (text): patient review
- **rating** (numerical): 10 star patient rating
- **date** (date): date of review entry
- **usefulCount** (numerical): number of users who found review useful

The data comes already split into train (75%) and test (25%).

3 The Methodological Approach

3.1 Preprocessing

The preprocess follows these steps:

1. Observation of NaN values: these represent only 0.05% of the total data (160 thousand records), so these can be eliminated.
2. Reviews are reduced to lowercase, and the leading and trailing quotes removed
3. Applied HTML decode of special characters (e.g. " ’ ")

4. URLs are replaced with the space character through "https?://\S+|www.\.\S+" regex
5. Emoji are replace with the space character
6. Replaced special characters related to cursor ("\n", "\r", "\t") with the space character
7. Replaced punctuation characters with the space character (ASCII symbols, except ".")
8. Replaced numbers through the regex "(\d\d*\.?)+" with the space character
9. Replaced ellipsis via regex "\.{2,}" with the dot character

In order to carry out each step starting from a coherent situation, at the end of each of these 8 steps, each review is brought to have a maximum of one space between words.

Then, for each review so transformed a process of sentence tokenization [3] is applied; to the obtained results it is then a applied process of word tokenization [3], giving as output for each review, a list of sentences, each of which is represented as a list of words.

In order to start reducing the vocabulary size, the different words are brought back to their basic form by applying the spacy library lemmatization process [4]. This proves to be better than the mild lemmatizer of nltk, while more tolerant than a stemming process, always through the nltk library (for example, whit the spacy lemmatization process the different verbal forms are reduced to a single form, while a term that has both an adjectival form and a nominal, is left unchanged depending on the case).

Thanks to a PoS tagging process, which is now possible on the obtained results, it is possible to keep exclusively the terms representing nouns (N*), adjectives (J*), verbs (V*) and adverbs (R*), which also includes negative forms such as ('not'). This choice is motivated by the maintenance of most of the meaning at the same time as a significant reduction in the vocabulary.

From this result, "stopwords" can be removed, they're already partially deleted in the previous steps, but some elements are still present (e.g. 'well', 'have', 'go'). Negative form stopwords (e.g. 'haven't', 'couldn't') are replaced with the string 'not_' which is concatenated to the next "non-stopword" term.

From the obtained reviews, two alternative versions are produced:

A first one, obtained through a Luhn pseudo-analysis [5], which considers the distribution of the logarithm (rounded to integer) of the frequencies of each word. The skewness of this distribution is observed, rounded and then used as an absolute cut-off value.

A second representation of the review is carried out considering the set of bigrams given by the grammatical constructs of the type "adjective + noun", "verb + adjective" [6], "adverb + adjective" [7], "adverb + verb" [7], "verb + adverb" [7].

Original	it helped me fall asleep and get a full night of sleep. it was not hard for me to stop taking it. i did not wake up feeling grogy.
Preprocessed	help fall asleep get full night sleep . not_hard stop take . not_wake feel grogy
Luhn Filtered	fall asleep full not_hard not_wake
Bigrams	full night | not_wake feel | get full | stop take | fall asleep

Table 3.1: Transformation applied to reviews

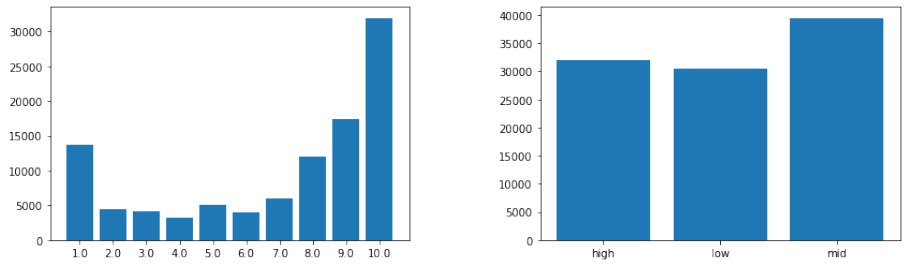
Original	i have had extremely positive effects from this drug but i do have to take prozac with it as it causes mood swings or depression with me.
Preprocessed	extremely positive effect drug take prozac cause mood swing depression
Luhn Filtered	extremely positive prozac cause swing
Bigrams	positive effect | mood swing | extremely positive | prozac cause | take prozac

Table 3.2: Transformation applied to reviews

3.2 Classification

For the classification task, different approaches and techniques have been tried, starting from the list of bigrams computed on each review as seen in Sec 3.1. The variable required to be classified is the review's rating in figure 3.2a, a number in the range of 1 to 10, which has been grouped into 3 classes:

low, mid and high; according to their frequency and values in an attempt to balance the class distribution in figure 3.2b.



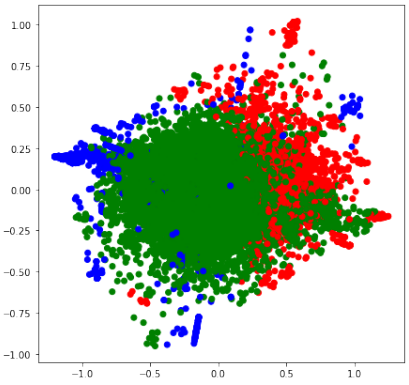
(a) Rating's distribution

(b) Rating grouped in classes

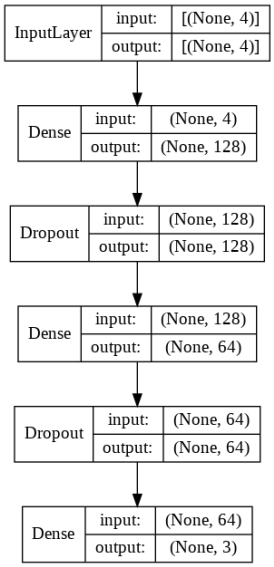
Figure 3.1

The first model tries to classify the rating based on the average sentiment of every word in the input. For this task it has been used the SenticNet dictionary [8] [9], which gathers sentiment polarity of over 180 thousand words. Each list of words is transformed into 4 features, obtained by the mean of all the word features, which are: introspection, temper, attitude and sensitivity. This new dataset created is then used as input for two different models: the Support Vector Classification, taken from the sklearn repository [10], and a simple Neural Network implementation in Keras. The NN model, in figure 3.2, is built with 4 inputs, one for each feature, followed by three dense layers of respectively 128 and 64 neurons with ReLU as an activation function and a dropout layer of 0.2. The last layer is then followed by a 3 neurons dense layer with Sigmoid activation for the final classification. Since the NN couldn't work with categorical data directly it was necessary to use one hot encoding on the three class variables.

The two models scored the same accuracy of 0.40, with the only difference being SVC took significantly more time to fit respect to the NN.



(a) SVC on test set PCAed



(b) Model's architecture

Figure 3.2

For this second try, a different input has been chosen. Each list of bigrams is joined to reconstruct the sentences, then a TF-IDF vectorization has been applied to construct a matrix of TF-IDF features. This has served as an input for a new NN model, in figure 3.4, made of two dense layers of 512 and 128 neurons each with a dropout of 0.2 and an ReLu as an activation function. Right after, the classification layer of 3 neurons with Sigmoid as an activation function.

The results show an improvement from the sentiment features, scoring 0.51 on accuracy, and open up for another implementation.

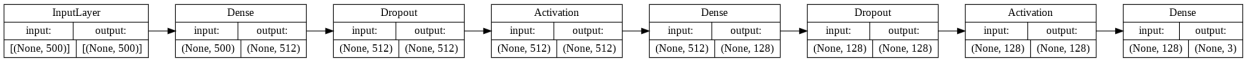


Figure 3.3: Model's architecture

Figure 3.4

On the third try, the NN model, in figure 3.6, is composed of two inputs, one is the same as the last attempt and the other is the embedding of the input sentence. This embedding is computed during the training cycle of the model via the Keras Embedding layer. From a list of embedding, one for

each word in the sentence, the mean embedding is calculated, given by the average of all, which is then concatenated with the TF-IDF features to be fed in the model. The model is then followed by three dense layers of 1024, 512 and 128 neurons each with a dropout rate of 0.3 and ReLU as an activation function. Same as before, the last layer is followed by a 3 neurons dense layer with Sigmoid activation for the final classification.

This last model overcomes its previous attempts, scoring an accuracy of 0.54 on both test and validation set.

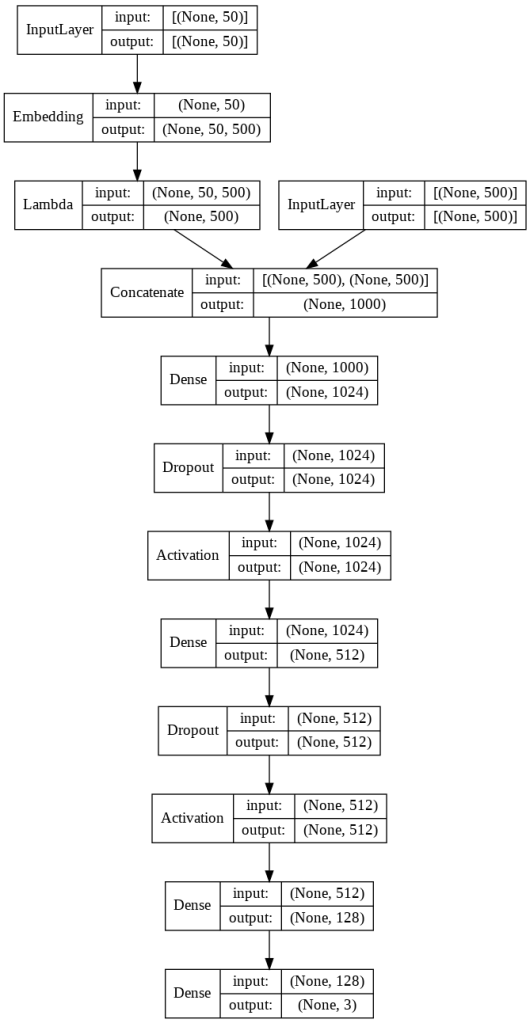


Figure 3.5: Model's architecture

Figure 3.6

3.3 Clustering

For clustering analysis, a Word2Vec representation of documents is used, which is preferred to the TF-IDF representation. This is because in the aggregation of the meanings related to drugs, it is preferred to consider the information given by the context rather than by the frequency of the present terms.

For the model realization the gensim library is used, in particular the Word2Vec package for the network [11], and the Phrases, Phraser packages for the realization of the bigrams [12]. The chosen model is SkipGram [13] and it is configured by placing the size of the carriers equal to 300 and it is trained with all reviews.

At this point, both the reviews in the basic form obtained by preprocessing and those derived through Luhn analysis and the bigram extraction are taken into consideration.

Each review is then transformed into a vector. In the case of reviews in basic form and those filtered by Luhn, the vector of a review is obtained from the average of the vectors of the words that compose it. In the case of the reviews given by the bigrams, first the vector of each bigram is obtained by concatenating the embedding of the two terms that compose it, then the average between these is performed in order to obtain a single representative vector.

The representation dimensional space is large, 300 dimensions in the case of reviews in basic form, 600 in the case of reviews given by bigrams. Two SVD reductions are then applied to reasonable spaces, one to 30 and the other to 50 dimensions.

With the obtained data, three different clustering methods are applied:

- **KMeans** [14]: even if the most appropriate distance with Word2Vec embeddings is typically a non-zero-match (for example a cosine rather than the euclidean distance) this algorithm is still considered as one of the most classic and "adaptable".
- **Agglomerative Clustering** [15]: hierarchical clustering of the agglomerative type. The matrix of the cosine distances between the vectors representing the reviews is used as input. Given the amount of observations, the complete method is used in order to calculating the linkage between clusters.

- **Gaussian Mixture** [16]: probabilistic model which assumes that the data are generated by a mixture of Gaussian distributions with unknown parameters. Assuming that the distribution of Word2Vec features follow a normal one, the idea is to use this method to grasp the greater diversity of context in reviews.

Each cluster method is performed by varying the number of clusters from 3 to 6, and by supplying as input both the reduced size data of 30 and 50 dimensions. For each of the executions, the silhouette internal measurement index and different external partitioning indices (e.g. Rand, Fowlkes Mallows) are collected.

Below the results obtained from the analysis conducted on the reference condition "Anxiety" are shown:

Method	**Words**	**SVD**	**k**	**Silhouette**	**Rand**	**Fowlkes Mallows**	**mean**
mixture	wl	30	5	0.09	0.62	0.13	0.28
mixture	bg	50	4	0.18	0.051	0.15	0.28
mixture	w	30	5	-0.002	0.73	0.12	0.28
mixture	bg	30	3	0.19	0.49	0.15	0.28
mixture	bg	50	3	0.21	0.46	0.15	0.28

Table 3.3

The method that turns out to be the best is Gaussian Mixture, by observing these results, and those obtained in other conditions, we opt to use bigrams, with dimensional space equal to 30 and number of clusters equal to 3.

On the subsequent conditions analysis, in order to give a more automatic process to the analysis, this method will always be used as it has proved to be the best compromise in stability between internal and external measurements and supervised sense degree of the results.

4 Results and Evaluation

4.1 Classification

Many models have been developed, with the help of increasingly complex and information-filled features. In fact, as can be seen in table 4.1 the evaluation

measures, in particular the accuracy, improve from model to model.

Model	Features	Accuracy
SVC	senticnet	0.40
Keras NN	senticnet	0.40
Keras NN	TF-IDF review	0.51
Keras NN	TF-IDF review + embedding	0.54

Table 4.1

Looking at precision, which is the number of classes correctly identified out of all the identified classes present, and recall, which is how many are correctly identified out of all the actual instances in that class, is easier to understand how the model performed.

	Precision	Recall	F1-Score
Low	0.57	0.56	0.57
Mid	0.51	0.57	0.54
High	0.51	0.45	0.48

Table 4.2

Specifically, analyzing the best model in Table 4.2, the low class stands out for its better scores compared to the other two classes. On the other hand, mid and high class have a lower precision but mid also have significantly higher recall than high. This may hint that the high class, despite being predicted the same amount of times as the mid class, is actually mixed up a lot with the other classes.

4.2 Clustering

In order to observe the informations of the obtained clusters, some results descriptors are collected As a first step, the association percentages of the reviews of each specific drug with respect to the total number of reviews is analyzed. This is made in order to observe which cluster best "describes" the drug.

Anxiety								
Cluster 1			Cluster 2			Cluster 3		
Drug	Cl. %	Total	Drug	Cl. %	Total	Drug	Cl. %	Total
Gabapentin	73.00%	175	Paroxetine	76.20%	97	Valium	63.20%	49
Xanax	67.60%	153	Desvenlafaxine	55.30%	54	Venlafaxine	47.60%	32
Diazepam	71.40%	140	Oxazepam	41.23%	18	Tramadol	37.60%	29

Table 4.3

It can be seen how the different drugs, even in large numbers, seem to have a reference cluster with a fair percentage, however, always by observing the values of the external validation measures, it is possible to understand how the method tends to group most of the reviews in a first cluster, then the others will represent the most dissimilar features: This could produce a slightly unbalanced distribution between the occurrences in each cluster.

Secondly, the most significant bigrams for each cluster are collected. To obtain them, the representative vector of each cluster was first created by averaging the vectors of the documents belonging to it. Then, for each cluster, among all the bigrams existing in that cluster, bigrams with the minimum distance from the relative representative vector are selected.

Finally, for completeness, the same procedure is also performed in order to observe which are the most representative words of the context of each cluster, among all terms existing in the vocabulary of the Word2Vec model.

The bigrams and words collected in the conditions Anxiety, Obesity and Opiate Dependence are shown below.

Anxiety					
Bigrams			Words		
Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
quickly relief	unaware chance	great not_pain	alone	crampy	massive
quickly induce	unpleasant effect	great help	impulse	overnight_pad	prevention
quickly feel	unable sleep	great sleep	obsessive	sleeve	effectively
quality life	nurse lie	great drug	decide	blast	migraine

Table 4.4

Obesity					
Bigrams			Words		
Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
fruit vegetable	upped dosage	quickly change	tremendously	dry_mouth	describe
breath anxiety	unfortunately take	quality life	not completely	struggle	health
bring depression	unfortunately not_cover	quick transition	not affect	deeply	considerabily
blur vision	urgent trip	quit point	huge	not$_h$*elp*	package

Table 4.5

Opiate Dependence					
Bigrams			Words		
Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
friend try	great success	understand addiction	side	not$_q$*uick*	heavy
full effect	great drug	probably save	remain	libido	heroin
frequent headache	free addiction	save marriage	suppress	childhood	balance
function addict	bump depression	save family	isolate	onset	nasal$_s$*pray*

Table 4.6

5 Conclusions

5.1 Classification

In the end, the classification task did not perform excellently, as long as low is properly identified, the mid and high classes are confused as shown, in table 4.2, by their lower precision and recall measures. A justification for this behavior may be the subjective view that the people have regarding the review's rating. Some can write terrific words but don't give the maximum score; on the contrary, other people will give a 10 without utilizing a much pleasant vocabulary. For this reason, a model which only knows about the review's text rather than the sentiment, can easily get confused.

5.2 Clustering

The clustering method obtains discrete results, the typical difficulty remains the interpretation of them.

From the conditions reported it is possible to see how in some clusters the representative bigrams are quite distinct in terms of positivity (cluster 2 and 3 in Anxiety, Table 4.4, and cluster 1 and 3 in Obesity, Table 4.5), while others only provide a possible context (cluster 1 and 3 in Opiate Dependence, Table 4.6). Generally the words extracted from the model to represent the clusters are aligned with the bigrams of the relative clusters and they can be interpreted in the same way.

However, in the case of bigrams, the effect of the concatenation between the words that compose it, is very evident (cluster 1 in Anxiety, Table 4.4, and cluster 3 in Opiate Dependence, Table 4.6), this method could therefore give too much advantage to bigrams composed of very similar words against bigrams in which both words are distinct but still belonging to a similar context.

6 Considerations

In the preprocess phase there are two major critical points: the first one concerns the management of the negation: the reduction from all its possible forms to a single term ('not_'), with the concatenation to the next word can help in the maintenance of general information, but replacing the negated forms (at least in adverbs and adjectives) with a term of opposite meaning could also allow a better differentiation of context.

Secondly, the transformation into bigrams through the used constructs helps to understand the type of reference to a drug, however, by breaking up the components of a sentence in this way it is possible that some meanings expressed in more than two terms are divided, with the consequent possibility of modifying the overall meaning. Searching through a larger number of constructs, even of different sizes, may be more effective.

Some improvements that can be made regarding the classification are the creation of an ensemble model, which gathers both the sentiment features and the textual features, TF-IDF and embeddings in particular. Another attempt may be using different dictionaries for sentiment analysis, which could bring more information. Furthermore, the classification problem may be treated as a regression problem, due to the nature of the review's rating variable which has been grouped into three classes to solve the class unbalanceness.

As in the classification, also in clustering, it could be useful to combine polarity features with those of context already provided by Word2Vec. An alternative methodology to concatenation in the construction of the vector of bigrams could also be useful, and then, a test of other clustering methods could reveal more suitable algorithms for this specific task.

Finally, the Drugs.com platform provides a list of possible side effects for each drug. With these it would be possible to perform a clustering analysis in which the similarity of the review to the description of the side effect is analyzed, in order to verify whether there is some partition by drug or condition that statistically reflects the clusters obtained.

Bibliography

[1] *Drug Review Dataset (Drugs.com) Data Set.* URL: https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29 (visited on 01/28/2022).

[2] *Drugs.com.* URL: https://www.drugs.com/support/about.html (visited on 01/28/2022).

[3] *NLTK: nltk.tokenize package.* URL: https://www.nltk.org/api/nltk.tokenize.html (visited on 01/31/2022).

[4] *Linguistic Features · spaCy Usage Documentation.* en. URL: https://spacy.io/usage/linguistic-features/ (visited on 01/31/2022).

[5] H. P. Luhn. *Luhn analysis. The automatic creation of literature abstracts.* en. 1958. (Visited on 02/13/2022).

[6] *Adjective and Verb Placement: Grammar Rules.* en. Mar. 2017. URL: https://www.grammarly.com/blog/adjective-and-verb-placement/ (visited on 02/02/2022).

[7] *Adverb placement | English Language Help Desk.* en-US. URL: https://site.uit.no/english/grammar/adverb-placement/ (visited on 02/02/2022).

[8] E Cambria et al. *SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis.* 2020. (Visited on 02/13/2022).

[9] *SenticNet. Helping machines to learn, leverage, love.* URL: https://sentic.net/downloads/ (visited on 02/13/2022).

[10] *C-Support Vector Classification.* URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (visited on 02/03/2022).

[11] *Gensim: topic modelling for humans.* en. URL: https://radimrehurek.com/gensim/models/word2vec.html (visited on 02/03/2022).

[12] *Gensim: topic modelling for humans.* en. URL: https://radimrehurek.com/gensim/models/phrases.html (visited on 02/03/2022).

[13] Tomas Mikolov et al.
Efficient Estimation of Word Representations in Vector Space. 2013. eprint: arXiv:1301.3781.

[14] *sklearn.cluster.KMeans*. en. URL: https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html (visited on 02/05/2022).

[15] *sklearn.cluster.AgglomerativeClustering*. en. URL: https://scikit-learn/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html (visited on 02/05/2022).

[16] *sklearn.mixture.GaussianMixture*. en. URL: https://scikit-learn/stable/modules/generated/sklearn.mixture.GaussianMixture.html (visited on 02/05/2022).